

AUTOMATED LECTURE VIDEO INDEXING WITH TEXT ANALYSIS AND MACHINE LEARNING

A Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Tayfun Tuna

May 2015

AUTOMATED LECTURE VIDEO INDEXING WITH TEXT ANALYSIS AND MACHINE LEARNING

Tayfun Tuna

APPROVED:

Jaspal Subhlok, Chairman
Dept. of Computer Science

Rakesh Verma
Dept. of Computer Science

Olin Johnson
Dept. of Computer Science

Shishir Shah
Dept. of Computer Science

Youmei Liu
College of Liberal Arts and Social Sciences

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I am very much grateful to my advisor, Dr. Jaspal Subhlok for his guidance, encouragement, and support during this work. He kept me motivated by his insightful suggestions for solving many problems, which would otherwise seem impossible to solve. I would not be able to complete my work in time without his guidance and encouragement.

I want to thank Dr. Shishir Shah, Dr. Rakesh Verma, Dr. Olin Johnson, and Dr. Youmei Liu for their co-operation during the various phases and their willingness to be a part of my thesis committee. I would like to express my deepest gratitude towards Dr. Christoph Eick, who gave me innumerable suggestions for machine learning. I want to thank Dr. Edgar Gabriel and Dr. Chad Wayne for providing valuable input data for this thesis. I am heartily thankful to Varun Varghese for his technical efforts on programming of text-based indexing algorithms.

Without the love and support of my family, it would have been hard to get my thesis done on time. I am forever indebted to my wife Naile Tuna and my daughter Rana Tuna who recently brought so much joy to our life.

AUTOMATED LECTURE VIDEO INDEXING WITH TEXT ANALYSIS AND MACHINE LEARNING

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Tayfun Tuna

May 2015

Abstract

Videos recorded during in-class teaching and made accessible online are a versatile resource on par with a textbook and the classroom itself. Nonetheless, the adoption of lecture videos has been limited, in large part due to the difficulty of quickly accessing the content of interest in a long video lecture. *Video indexing*, dividing the video into meaningful segments, can significantly improve the accessibility. In this work, we present automatic text-based approaches and machine learning for indexing lecture videos to provide topic-based segmentation.

Various text-based indexing algorithms were developed to identify topic transition in video. The indexing algorithms merge neighboring video segments with high text similarity to form topic segments which are represented by *index points*. In general, it is not clear which feature in a video slide is important for detecting topic change. Therefore, we propose another video indexing approach using machine learning which can use all possible features such as the number of words in a slide, n-grams, title or text with large font size. Among the state of the art machine learning algorithms, ensemble models such as Random Forest and Bagging were found efficient and practical to use. They also provide probability distributions which enables the user to choose a desired number of index points.

Evaluation was done on a set of twenty-five lecture videos from courses in Computer Science, Biology, and Earth and Atmospheric Science. The ground truth is established by asking the lecture instructor to manually identify topic transitions in the video. Information gain experiment with machine learning shows that the words with large font size, the words that appear in the video for the first time, and n-gram frequency differences between video slides are important features for identifying the

topic transitions in a lecture video. Experimental results shows that text-based indexing provides significant improvement over non-text-based approach and indexing with machine learning provides approximately 80% indexing accuracy on average. An important observation was that, there are significant differences when the topics are manually identified by multiple users who are very familiar with the content. Although further enhancements could improve the performance of video indexing, the performance gains are not expected to reach the ideal output because of the uncertain nature of the ground truth.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Objective and Research Questions	5
1.3	Dissertation Outline	7
2	Background: ICS Videos Project	8
2.1	Indexing	9
2.2	Keyword Search	10
2.3	Captioning	10
2.4	ICS Video Player	11
2.4.1	Index Panel	12
2.4.2	Search Display	12
2.4.3	Caption Display	12
3	Related Work	13
3.1	Delivering Course Work Online by Lecture Videos	13
3.2	Text Extraction and OCR Engines for Videos	14
3.3	Video Indexing	15
3.4	Text Segmentation	17
3.5	Using Machine Learning	18

4	Lecture Video Indexing Framework	19
4.1	Identifying Transition Points	19
4.2	Identifying Index Points	20
4.3	Uniform Indexing	22
5	OCR Text Extraction	26
5.1	Text Extraction	26
5.2	Image Enhancement	28
5.2.1	Segmentation	28
5.2.2	Color Inversion	30
5.3	Evaluation	33
6	Ground Truth Creation and Accuracy Metrics	35
6.1	Ground Truth and Rating of Transition Points	36
6.2	Indexing Score Calculation	41
6.2.1	2-Point Metric	41
6.2.2	4-Point Metric	42
6.2.3	Sorting Metric	45
7	Text-based Indexing	48
7.1	Fixed Grouping Text-based Indexing Algorithm	48
7.2	Linear-weighted Text-based Indexing Algorithm	51
7.3	Non-linear Weighted Text-based Indexing Algorithm	55
7.4	Boundary-based Text-based Indexing Algorithm	58
7.5	Term Frequency-Inverse Document Frequency (TF-IDF) Optimization	60
7.6	Evaluation	62
8	Indexing by Machine Learning	63

8.1	Objective of Machine Learning for Video Indexing	64
8.2	Dataset and Creating Feature Vectors	65
8.3	Handling 4-Level Input and 2-Level Output	68
8.4	Need for Number of Index Points Parameter	74
8.5	Choosing Machine Learning Algorithms: Ensemble Models	78
8.6	Attribute Importance by Information Gain	81
9	Evaluation and Experimental Results	85
9.1	Indexing Accuracy	85
9.2	Survey Results	94
10	Limitations and Discussions	99
10.1	Index Points Which are not Detected by Any Algorithm	101
10.1.1	Outline or Summary Slide in a Hierarchical Lecture Organization	102
10.1.2	Similar to Previous Topic: Ambiguous	102
10.1.3	Animations and Slide Transition Effects	103
10.1.4	Slide Without Enough Text for Topic Information	104
10.2	Other Common Errors	105
10.2.1	Slide Revisit	105
10.2.2	OCR Error and Hand Writings	106
10.2.3	Image Captions or Irrelevant Text	107
11	Conclusion	109
11.1	Summary of Key Contributions	110
11.2	Future Work	111
	Bibliography	113

List of Figures

1.1	Student ratings of studying resources	3
1.2	Student responses for the reasons of using lecture videos	3
2.1	ICS Videos Framework: Indexing analyses frames, extract images. Search module enhances images and applies OCR text extraction. Video indexing uses the extracted images and texts to create index points. The results are stored in database for ICS Video Player . . .	9
2.2	A snapshot of ICS Video Player	11
4.1	Transition point in video: third frame is a new transition points. . . .	20
4.2	Indexing framework steps: 1) Unique video frames detected by RGB Color difference and transition points are defined. 2)Index points representing different concepts are selected among the transition points.	21
4.3	Uniform indexing algorithm steps: in each step shortest segment is merged to immediate left or right based on the text similarity. Therefore, 1,4,5,7, and 8 selected as index points.	25
5.1	Example of ICS video frames which is a challenge for OCR	27
5.2	Dilation effect on an image: Dilation joins the small objects (characters) and fills the small holes; text is converted to square objects . . .	29
5.3	Segmentation and enlargement of text	31
5.4	Inversion example: Original image and color inverted images. It is an open question that which image is more readable and which image will have better OCR results.	32
5.5	Search accuracy rate of OCR tools	34

6.1	Interface for ground truth index point creation: 3-Definitely index points, 2-Probably index point, 1-Probably not an index point, 0-Definitely not an index point	38
6.2	Coursera interface to access the videos: each video is divided into segments	39
6.3	Scoring algorithms with sorting metric	46
6.4	Applying sorting metric to score algorithms with ground truth	47
7.1	Fixed grouping algorithm example: Shortest segment compared to left and right group of neighbor segments and how it is merged based on similarity	50
7.2	An example of reason for fixed grouping of segments	51
7.3	Linear weight reduction of the transition points	54
7.4	Weight decay for various half-life	57
7.5	Non-linear weight reduction	58
7.6	Boundary-based index point selection	59
7.7	Indexing (4-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in the ground truth	62
8.1	Creating dataset for machine learning from ground-truth table	68
8.2	4-Level input and 2-level output challenge in machine learning	69
8.3	Possible strategies to handle 4-level input and 2-level output in machine learning	70
8.4	Dividing dataset to train and test to find the best approach for handling 4-level input and 2-level output	72
8.5	Processing traindata and testdata to define best approach to handle 4-level input and 2-level output	73
8.6	Experiment result on train and test dataset for different approaches with various machine learning algorithms: converting to binary performs better in all algorithms	73
8.7	Can we do video indexing by using machine learning without providing number of index points parameter	75

8.8	Actual number of index points vs findings of machine learning as index points	75
8.9	Correct strategy to apply machine learning for video indexing: Number of index points should be provided to machine learning and post processing should be done to select desired number of index points. .	76
8.10	Seven steps for indexing video with a machine learning algorithm . .	77
8.11	10-Fold Cross Validation results of different machine learning algorithms: high accuracy with Ensemble models AdaboostingM1, Bagging and RandomForest.	79
8.12	10-Fold Cross Validation results of different metrics for ensemble models: AdaboostingM1 has very low true positive rate and very low precision	80
8.13	Dataset is imbalanced: number of <i>not index point</i> is much higher than <i>index points</i>	81
9.1	Indexing (4-point metric) average accuracy for 25 videos; index point per 6 minutes	86
9.2	Indexing (4-point metric) average accuracy for 25 videos; index point per 8 minutes	87
9.3	Indexing (4-point metric) average accuracy for 25 videos; index point per 10 minutes	88
9.4	Indexing (4-point metric) average accuracy for 25 videos; index point per 12 minutes	89
9.5	Indexing (4-point metric) accuracy for index point per different time interval for all videos	90
9.6	Indexing (4-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in ground truth	91
9.7	Indexing (2-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in ground truth	92
9.8	Number of transition points and number of index points based on video duration: not linearly correlated	93
9.9	Sorting metric indexing score of TB-Boundary, ML-RandomForest, and ML-Bagging	93

9.10	Value of video indexing with first set of questions	94
9.11	Value of video indexing with second set of questions	95
9.12	Value of search with first set of questions	96
9.13	Value of search with second set of questions	97
10.1	Different ground truths and indexing (4-point metric) average accuracy of 10 videos: each person marks differently	100
10.2	Distribution of cause of errors for index points that are not detected by any text-based and machine learning algorithm	101
10.3	Outline slide in a hierarchical lecture organization	102
10.4	Consecutive slides have very similar topics	103
10.5	Animation and slide transition effect: next slide adds an additional line to previous one	104
10.6	Slides have only images or have very low amount of text do not provide enough information for algorithms to correctly find index points . . .	105
10.7	Going back to previous slides in lecture organization causes a break in the similarity between the segments	106
10.8	OCR errors on hand writing: false detections mislead the algorithm to treat same slide as different slide	107
10.9	Texts which are irrelevant to the topic in image captions cause algorithms to produce false index point	108

List of Tables

6.1	List of Department and Courses used for indexing experiment	36
6.2	Ground Truth for 25 Lecture Videos	40
6.3	2-Point scale accuracy matrix	42
6.4	Possible indexing scores for a transition point with 4-Point scale . . .	44
8.1	10-Fold Cross Validation results of different ensemble models: true positive, true negative, false positive, false negative	80
8.2	The most important 50 features based on information gain	82
8.3	The least important 50 features based on information gain	83
8.4	All features created for machine learning indexing	84
9.1	Survey Map: video usage, problems, indexing and search value	98
10.1	Distribution of undetected index points: false negatives	101

Chapter 1

Introduction

Video is gaining popularity as a learning resource. Video recordings of classroom lectures are often made available as additional material for a conventional course, as the core of a distance/hybrid learning course, or posted publicly for community learning. Lecture videos are posted on a large scale on portals such as MIT OpenCourseware and Apples iTunes University. In recent years MOOCs (Massive open online courses) driven by video and other features have emerged as a potential disruptive technology for delivery of education. The most important virtue of a recorded lecture video is that it is anytime anywhere while approximating the classroom experience. A critical weakness of the video format is the inability to quickly access a topic of interest when video is used for reference.

1.1 Motivation

The motivation for this dissertation comes from the desire to provide meaningful and topic-based accessibility to lecture videos so that the students can easily access and review the topic of their choice quickly and efficiently. Studies show that videos are very important as educational material and they are used mostly for review purposes [6]. Figures 1.1 and 1.2 show the survey results made in between 2010-2011. Data were collected from 2,394 students at the end of each of five semesters between spring 2009 and spring 2011 at the University of Houston. Students were asked to rate the importance of lecture videos in comparison to other resources made available by faculty, including professors' lecture notes, students' own notes, and the textbook assigned for each class. As Figure 1.1 shows, in relation to getting the grade they wanted for the class, students gave the second highest rating to lecture videos, with 64 percent of students reporting that this resource was "very important". As shown in Figure 1.2, the most commonly reported use of lecture videos was to review for a test or assignment (77 percent) or review difficult concepts (77 percent). Nonetheless, there is problem in video format for quickly accessing the content.

Textbooks are organized by chapters and sections based on topics and subtopics. A reader can immediately find a chapter in the book from the table of contents or find locations where a topic is discussed based on keywords from the index at the end of textbooks. In contrast, accessing the content of interest in a lecture video is not easy because there are no table of contents or index sections. Often the only way to find a topic of interest in a video is by scrolling the video from the beginning,

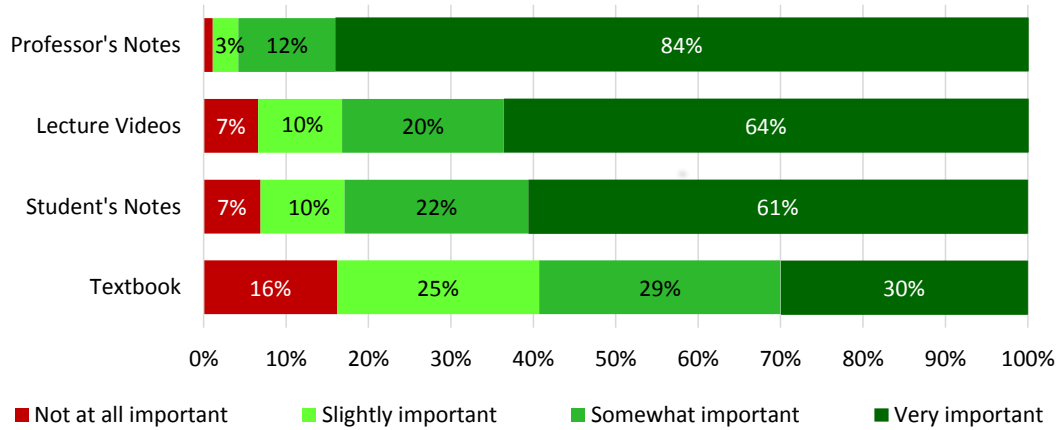


Figure 1.1: Student ratings of studying resources

which can be time consuming and frustrating, especially for long videos.

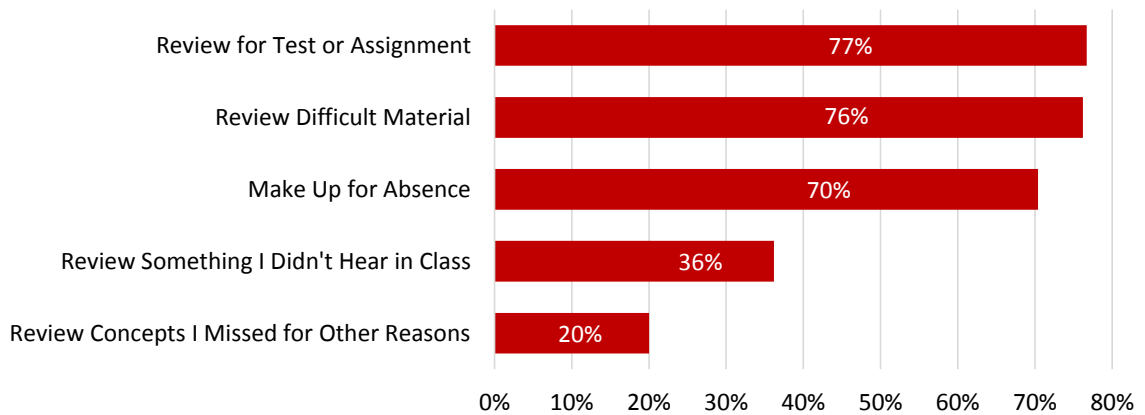


Figure 1.2: Student responses for the reasons of using lecture videos

Video indexing aims to overcome this challenge by dividing a lecture video into segments that contain different topics or subtopics. The beginning of each segment is called *index point* that is visually represented by the image at the beginning of that time segment. A user can visually see the location of various topics and subtopics with these index point images and can easily access the content of interest or switch

between topics by clicking on the index point images. It is possible to manually annotate or segment the topics in a video lecture by the instructor. However, manual segmentation is a time-consuming and labor-intensive process because a typical lecture video duration can be ninety minutes or more. Manually segmenting the entire duration thus becomes an expensive operation. However, automatic segmenting process provides a faster and more efficient solution. The video segmenting based on time duration or by detecting the scene changes within the video does not guarantee topic-based segmentation. A video comprises a sequence of images and a typical lecture video contains the text information within these images. Extracting and analyzing the text for topic changes can, in principle, provide more accurate topic segmentation results than non-text-based methods because text is a better indicator of topics.

One of the common techniques with text analysis used for finding the topic change is checking where the mix of words changes significantly. The core of this approach is that topics are explained by words and if the words change in the text, the topic will change. Text can be represented by frequency of word vectors and the topic shift can be measured by the angle between these vectors. We can examine the topic change in the video by checking the word shift with this approach and create a text segmentation algorithm for lecture videos.

Lecture videos are composed of presentation slides that have some unique characteristic in terms of text which may affect the topic change. For example, words with different font size and different font color has a value in a slide. For a topic change, change of words having large font size may be a better indicator than the change

of words having regular font size. Or each slide text is shown for a certain amount of time which maybe a sign for importance of the text: if the text is shown in the video for longer time this text maybe more important than others. It may even be a sign for a new topic explanation since new concept explanation requires more time than continuing with a concept. Thus, it is not clear what features are important in a video text for detecting topic change. It is also not feasible for us to create text segmentation algorithms manually, considering all these aspects. Instead, we can use machine learning algorithms for our purpose and discover the relationship of all these aspects. Since machine learning algorithm uses statistical and mathematical approaches in direct samples of problem, these algorithms can deal with as many features as provided.

1.2 Objective and Research Questions

In this dissertation, we aim to analyze the text for finding the topic transition in videos so that we can divide lecture into segments that contain different topics or subtopics. Each topic is called an *index point* that is visually represented by the image at the beginning of that time segment. Users should be able to access those index points and switch between the index points easily. Thus, the number of index points should be in a reasonable range for navigation.

This dissertation is motivated by two main research questions: (1) How can we use the text information to find the index points and how can we create a text-based indexing algorithm? (2) How can we use state of the art machine learning algorithms

to find the index point?

These questions require further examination through these queries:

- How can we extract the images and text from video?
- Can we use the current Optical Character Recognition (OCR) tools for extracting the text from video images? What is the accuracy of current OCR tools and what can be done to improve the output of OCR tools?
- How can we measure the text similarity and what type of approach should be followed to develop an indexing algorithm?
- Which features are important to detect topic change?
- How can we create a ground truth to evaluate text-based indexing algorithm?
- How can we create a dataset for machine learning?
- Which machine learning algorithms can be used and how we can we apply them for video indexing?
- Which approach is better, a text-based indexing algorithm or indexing by machine learning?
- What are the limitations and challenges of automated tools for detecting topic change in a lecture video?

1.3 Dissertation Outline

The organization of the remainder of this dissertation is as follows. Chapter 2 presents the background of this work with Indexed Captioned Searchable (ICS) videos project and its components. Chapter 3 is the literature review of existing approaches for the system that builds up the entire ICS framework: using lecture videos to deliver coursework online, text extraction from videos, using text information for video indexing, and document segmentation and using machine learning for video indexing. Chapter 4 explains the video indexing framework and outlines the steps for finding the index points. In Chapter 4, text extraction by OCR tools along with the type of image enhancements. In Chapter 6, the ground creation for evaluation purposes and accuracy metrics used in this dissertation are explained. Chapter 7 introduces a text-based indexing algorithm and various enhancements. Chapter 8 investigates the procedures for how to use machine learning algorithms. In the following chapter, experimental results done on set of videos presented. In Chapter 10, error analysis is done and limitations and challenges for automated lecture video indexing are discussed. Finally, the last Chapter concludes and summarizes the dissertation highlights and lists the key contributions.

Chapter 2

Background: ICS Videos Project

The ICS Videos Project stands for indexed, captioned, and searchable videos. This project aims to make the lecture videos easily accessible by providing indexing, keyword search, and captioning. This thesis aims to enhance the indexing mechanism in ICS Videos Project by providing topic-based indexing based on the text content in the video. The key components are the video indexer, the captioning module, the keyword search mechanism and the custom video player. The workflow for the ICS Videos Framework is depicted in Figure 2.1. Once a video is uploaded, the images on the video are extracted selectively. Images are enhanced by image transformations so that text on the images can be extracted by OCR. The text on the images is analyzed by the indexing module. The keywords and their location, along with the index points are stored in the video player database. Content and time location of captions are also included in the database when available. The video player accesses the database on demand to support indexing, search and captions.

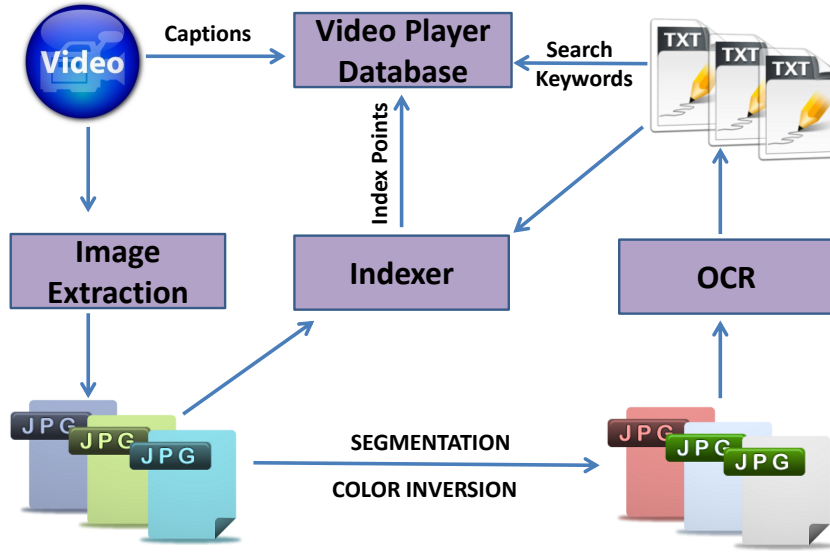


Figure 2.1: ICS Videos Framework: Indexing analyses frames, extract images. Search module enhances images and applies OCR text extraction. Video indexing uses the extracted images and texts to create index points. The results are stored in database for ICS Video Player

2.1 Indexing

Indexing is the task of dividing a lecture video into segments that represent different sub-topics. This task is accomplished by first identifying all transition points where the scene in the video changes significantly. Next, a subset of these transition points is selected as index points, which represent the beginning of the video segments as presented to the user.

Detection of transition points is based on the color comparison of successive frames in the video. The RGB (Red, Green, Blue) values of corresponding pixels in the two images determine the similarity between the images. Since comparing pairs of all successive frames in a video is rather inefficient and time consuming, binary

search method and jumping interval is used to speed up the process. Image difference between successive *transition points* forms the criteria for *index point* selection in the previous framework. Evaluations of this method indicated that the selected *transition points* aligned perfectly with the scene transitions within the video; however the *index points* were acceptable but not always represent a topic change.

2.2 Keyword Search

In keyword search, all the video segments containing the keyword are identified and presented to the user. The procedure to support keyword search is as follows. The indexer creates the video segments as well as the transition point frames. Text on these frames is detected by OCR module and stored in a database. The ICS Video Player loads the keywords from this database along with the corresponding video. When user searches for a keyword, the player presents a series of index points that allow the user to navigate to the corresponding video segments.

2.3 Captioning

In captioning, the caption box in the ICS video player presents the audio stream in the lecture video as synchronized text block. This audio typically consists of the instructor's voice as well as student interactions. ICS framework can generate the captions automatically, but a certain degree of manual correction is required because of the limitations of the speech to text conversion technology.

2.4 ICS Video Player

ICS Video player is an HTML5 based player capable of playing streaming video over the internet. The player consists of a playback component, index panel, keyword search box, and a transcript display panel. Figure 2.2 shows a snapshot of the ICS video player highlighting its key features.

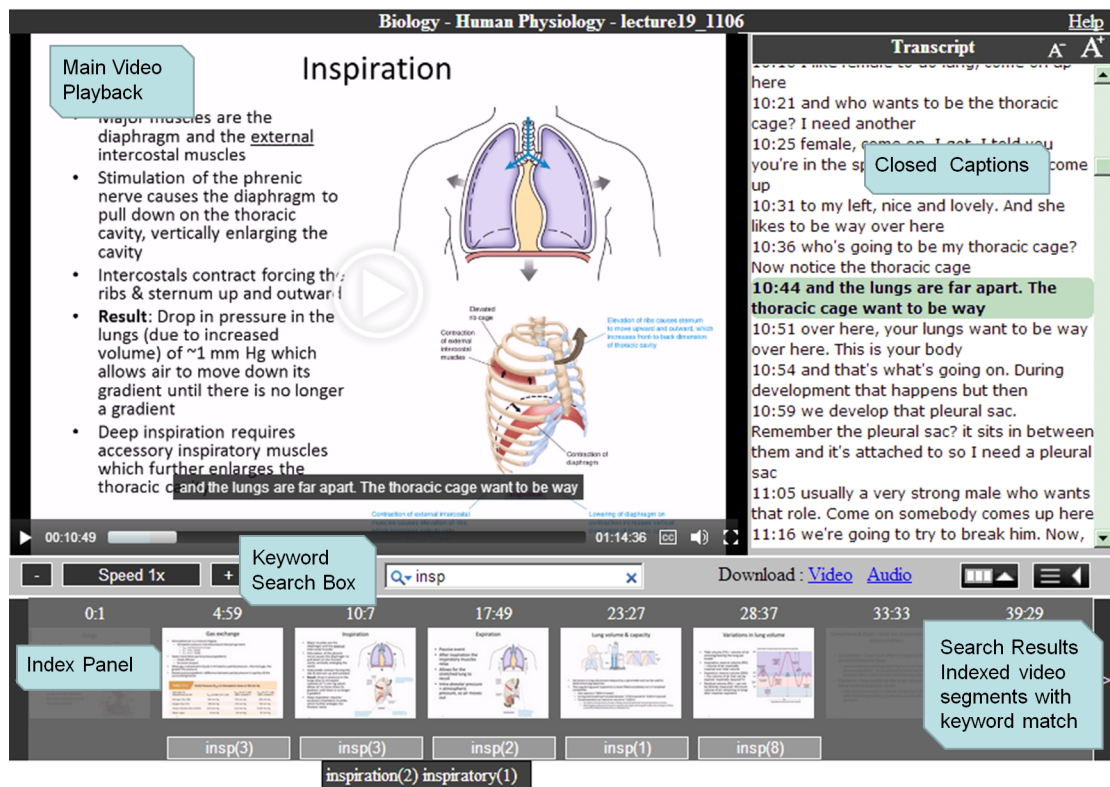


Figure 2.2: A snapshot of ICS Video Player

2.4.1 Index Panel

An index panel is situated on the bottom of the player, as shown in Figure 2.2. Index points are listed vertically in the index panel as the thumbnail images. These images represent the starting frame of the topic in the video. The text on top of the images displays the start time of the respective Index Point so that the users can easily understand the spacing of the topics within the video.

2.4.2 Search Display

A keyword search box lies in between the index panel and the main video playback area. The user types the keywords in the search box. The index panel retains all the segments that contain the keyword and deactivates the remaining segments or index. The area below each thumbnail image display the matching keyword as well as the number of matches. This helps to convey the importance of a segment relative to the search keyword.

2.4.3 Caption Display

An overlay at the bottom of the video displays the captions, if available. A separate panel on the right-hand side of the video player displays the complete transcript. Based on the playback, the full transcript section highlights the corresponding caption and its position automatically updated on the screen. Scrolling allows reading of the complete transcript.

Chapter 3

Related Work

This chapter reviews and discusses performance, relative merits and limitations of existing approaches for each step in ICS Video Framework and several approaches for analysis of video usage and video indexing in the literature.

3.1 Delivering Course Work Online by Lecture Videos

The idea of automatically capturing videos of class lectures, conference presentations, and talks followed by means of and presenting / distributing them as videos has been around for a long time [1, 46, 32, 3]. These videos are mostly recorded by camera(s) operated by professionals or are edited from footage captured from cameras which are installed in the lecture/presentation rooms [8]. As the number of videos increased,

attempts have been done to automatically index the videos or create digital library for better information retrievals [23, 29]. Or to increase the accessibility and usability, these videos are manually edited [26, 15]. The project Lectern II [26], employs the touch-sensitive screen technology to build a digital desk, which is shown to be able to effectively support and transparently capture the standard classroom lecturing activity. Recorded lectures can be edited and automatically uploaded to a Web server and then viewed by students via standard streaming player. But editing video is done manually. There exists a related technology known as Hypervideo which can synchronize content inside a video with annotations and hyperlinks [33]. Hypervideo allows a user to navigate between video chunks using these annotations and hyperlinks but one still has to manually put annotations and hyperlinks in the Hypervideo to index it. An interactive online learning system Coursera is using segmented videos which are manually edited and partially recorded. The approach in this work is different from these state of the art systems by being fully automatic and independent of any hardware or presentation technology. Advantages of ICS videos include excellent resolution because the video consists of screen captures on the PC itself and very low production cost as no camera or operator is involved.

3.2 Text Extraction and OCR Engines for Videos

Different techniques have been used to index the videos and to make search inside the video feature available: using text on slides [8, 35, 47, 48, 49, 45] and texts obtained by speech recognition tool [42]. Authors of [8] proposed a fully automatic

method for summarizing and indexing unstructured presentation videos based on text extracted from the projected slides. They use changes of text in the slides as a means to segment the video into semantic shots. Once text regions are detected within key frames, a novel binarization algorithm, Local Adaptive Otsu (LOA), is employed to deal with the low quality of video scene text. We are inspired by this work by its application of threshold to images and its use of the Tesseract OCR tool. Authors of [29], worked on Automatic Video Text Localization and Recognition for content-based video indexing for sports applications using multi-modal approach. They used segmentation by using dilation methods for localizing. The method for segmentation in this work is inspired by this work. The work of [2] provides a keyword search inside the lecture video which is one of the few closest work done to this work which lacks of search accuracy and lacks of showing the keywords that are found.

3.3 Video Indexing

Automatic video annotation or indexing involves the detection of key frames or labels that indicate change of content in a video [22, 21, 36, 38]. State of the art in computer vision, pattern recognition, and image processing has enabled automatic indexing based on a variety of content cues. The work by Davis [14] provides a low degree of automation through the use of high level ontological categories like action, time, space, etc. A multitude of methods have also been developed that use low-level image properties such as color, texture, etc. These techniques use similarity of image

properties to group contiguous video segments and provide reasonable automation while lacking the ability to provide semantics to grouped segments [39, 37, 5]. In [12] methods for segmenting the video using signature, requires to use an input from instructor. And environment semantic analysis on frames also is not applicable to our work since detecting the scene will not provide much information for classroom recorded lecture. Using a speech recognition tool to find the words and time intervals and indexing them by these intervals is another implementation of video indexing [42]. Some selected semantic words such as *example*, *exercise* are used for search to help students to navigate in videos. Our approach on text based indexing also uses time interval information. But it is not per word but per slide and we cannot limit the search terms to specific words. Because this will require providing a different set of specific words for each videos while it is unknown what the user interested in each lecture video content.

Lecture video segmentation based on the linguistic features of text [30] is very similar to the work that is presented. Comparing the text segments extracted from the lecture videos for similarity determines the text boundaries, where the similarity is low. However, a dictionary-based approach identifies different types of words (part of speech) such as nouns, verbs, pronouns, etc., these are separately represented as features. The similarity calculation is only between selected individual features. Human supervision is required in this dictionary-based approach for customizing the dictionary for a particular video subject. This thesis differs from the previous research, as the video indexing method is unsupervised and considers all the words irrespective of the kind of word or feature.

3.4 Text Segmentation

There are many works on text segmentation techniques implemented on different literature corpus. The goal of these methods is to measure the gap between text by calculating the angle between vectors. A mathematical measure such as the cosine similarity is used [25]. Hearst[24] introduced the topic based text segmentation algorithm TextTiling. It segments texts in linear time by calculating the similarity between two blocks of words based on the cosine similarity. This algorithm was not compatible with the lecture video text due to the low amount of text in slides and uncertainty of block size. Another similar approach is Choi's C99 algorithm [13]. C99 uses the similarity matrix to build local ranking of proximity between sentences. The more similar to their neighbors the sentences are, the higher their ranks. The lowest rank in the new built ranking matrix shows the boundary between the two main parts of the text. These two parts are then considered as two independent texts, and the algorithm is applied on each part. The algorithm stop when the lowest rank detected is the last sentence of the analyzed part of the text. This technique cannot be applied in video slide text. Because in video, boundaries of texts are not sentences, but slides and some slides have very low amount of text comparing to others which makes it hard to create boundaries based on slides.

3.5 Using Machine Learning

Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed [43]. There are many works, mostly used in computer vision applications, that create features for videos and use machine learning for different purposes such as synchronization of video scenes [19], finding the scene changes in dynamic video [18], or finding which video belongs to which category [11, 40]. Applications using the text information in video for machine learning is quite limited and one the closest work to our work is finding transition slides in a lecture video [31] which is still not related to topical video segmentation. To the best of our knowledge, we could not find any similar work using machine learning and text for topical segmentation of the lecture videos.

Chapter 4

Lecture Video Indexing Framework

Indexing is the task of dividing a lecture video into segments that contain different topics/subtopics. This task is accomplished by 2 steps. The first step is identifying all transition points, i.e., places where the scene on the video changes significantly. Secondly, a subset of these transition points are selected as index points which are the starting points of video segments presented to the user.

4.1 Identifying Transition Points

In Figure 4.1, an example of a transition point is shown. Among the 5 consecutive frames, the third one is the transition point and detecting this point is done by comparing the differences of image pixels. Corresponding pixels in successive frames

are considered different if they differ by a minimum RGB threshold when the RGB values of the pixels are compared. Successive frames constitute a transition point if the fraction of pixels that are different based on the RGB criteria exceeds a minimum threshold that we refer to as the transition point threshold. The reason for using thresholds to identify transition points is that frames corresponding to the same scene in practice (e.g., exactly the same viewgraph) also have minor differences in the RGB spectrum that must be ignored to avoid false transition points. The threshold values are chosen empirically after evaluation of a large number of diverse lectures. A value of 10% was selected for both RGB threshold and transition point threshold for the system used in this work. Details are explained in these works [7, 28]

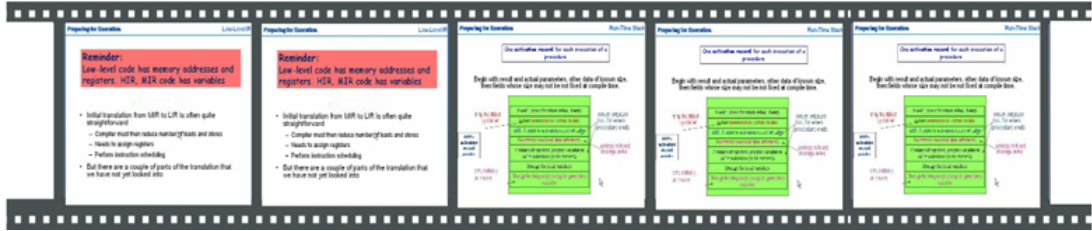


Figure 4.1: Transition point in video: third frame is a new transition points.

4.2 Identifying Index Points

Index point represents the start of a topic and is a subset of the transition points in a video. Consecutive transition points that are part of the same topic are grouped together to form a continuous segment, represented by the index point. Figure 4.2 shows the selection of the most suitable index points from a list of transition points. A video may have a large number of such transition points, e.g., over 100 transition

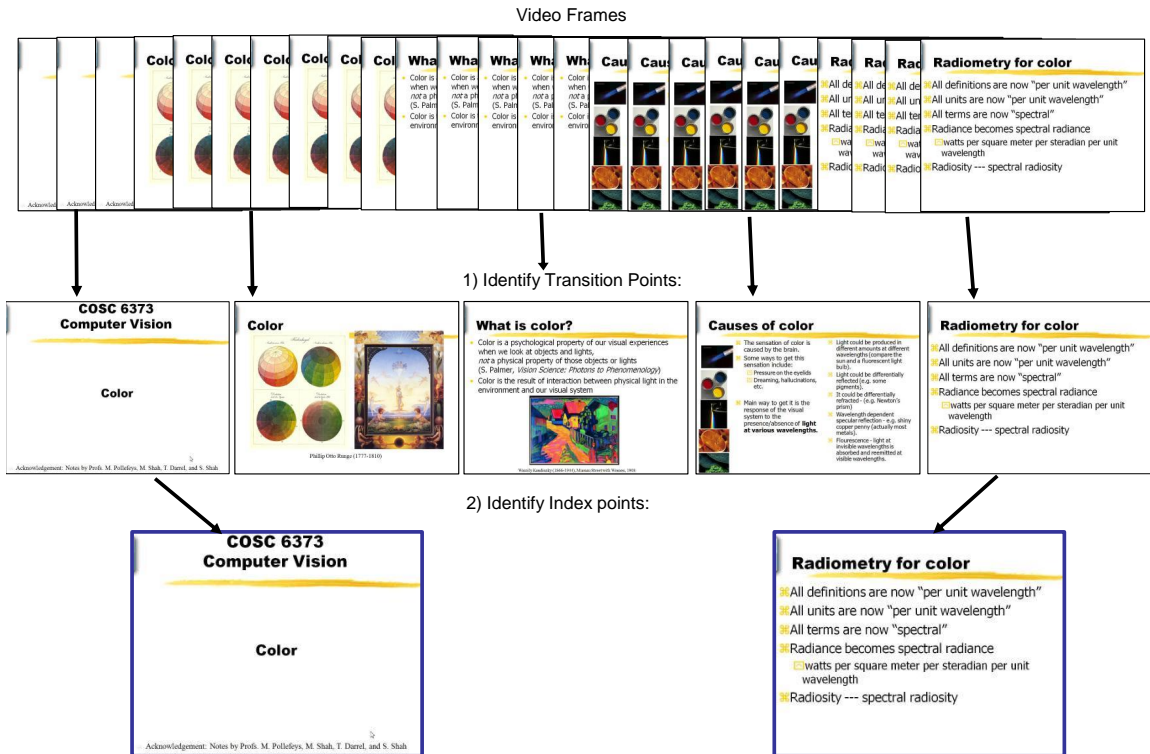


Figure 4.2: Indexing framework steps: 1) Unique video frames detected by RGB Color difference and transition points are defined. 2) Index points representing different concepts are selected among the transition points.

points, is not unusual when an instructor is writing on a Tablet PC screen frequently. The goal is to identify a smaller number of index points that are more meaningful and easier to use. To identify the index points we can use different approaches:

- Uniform indexing
- Text-based indexing
- Machine learning indexing

The text-based and machine learning indexing are expected to perform better than the non-text-based, baseline algorithm. And they will be discussed in detail in the following sections.

4.3 Uniform Indexing

The Uniform indexing algorithm is a non-text-based algorithm for indexing the video and forms the baseline for comparing the performance of other text-based algorithms. This algorithm is based on the time duration of the transition segment. The Uniform indexing algorithm aims to uniformly distribute the index points throughout the video where the scene changes occur. The algorithm is explained as follows.

Data: A list of transition points ;

Required number of index points (N)

Result: N index points which is a subset of transition points

repeat

 Select transition segment with smallest duration;

 Merge with the smallest neighbor. In case of tie, merge with the left;

until *Number of transition segments == Required number of index points;*

Algorithm 1: Uniform indexing algorithm

The goal of the indexing process is to segment the video according to the individual topics such that each index point represents a topic within the video. The initial phase segments the video into transition segments. Indexing phase involves comparing the similarity of transition segments in the input list with the left and right segments for similarity. The segment is merged with its immediate left or right neighbor depending on which side has a greater similarity value.

The algorithm advances by selecting the transition segment with the shortest duration and merges it with the neighbor that is of shorter duration. The merging process is a conceptual process that ignores the boundary between the two segments and considers the two segments as a single segment. The duration of the merged segment is the total duration of the merged segments. The resulting text content of the merged segment is the sum of the text of the two merged segments. A topic typically consists of several segments. Generally, a segment of small duration is part of a topic and does not form the entire topic. It is possible to split the video into segments of equal duration to achieve a uniform distribution of index points.

However, the resulting index points do not coincide with the scene transition, where the viewgraphs change. For these reasons, the algorithm always selects the segment with the smallest duration and merges with the most suitable neighbor.

Figure 4.3 gives a pictorial example of Uniform indexing algorithm. The numbered rectangular blocks represent the transition segments in the video. Required number of index points is set to 5. In this example, at first iteration, the smallest segment 6 merges with the smaller of its neighbor, segment 5. In the next iteration, the smallest segment is 9 and merged to its smallest neighbor, segment 8. This processes continues until the number of segments are equal to required number of index points.

Uniform indexing algorithm does not provide topic based indexing. Instead, the algorithm distributes the index points at approximately uniform interval of time. This algorithm forms the baseline for comparing the performance of text-based indexing algorithms.

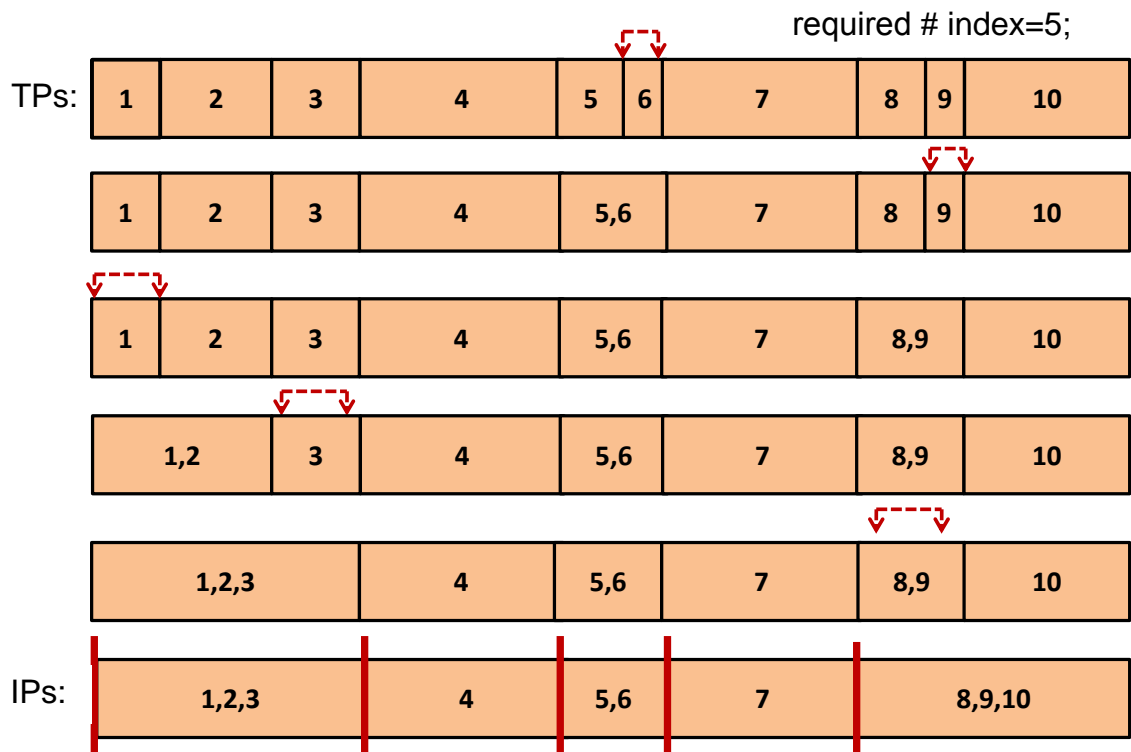


Figure 4.3: Uniform indexing algorithm steps: in each step shortest segment is merged to immediate left or right based on the text similarity. Therefore, 1,4,5,7, and 8 selected as index points.

Chapter 5

OCR Text Extraction

5.1 Text Extraction

Text-based video indexing and keyword search require that the text contained in the video frames be identified. In this section, the methodology for recognizing text in a video frame is presented. Clearly, it is not necessary to recognize the text on every frame in a video as sequences of video frames typically have identical text. Selection of frames for text recognition is part of our methodology for identification of index points discussed in next Chapter.

Recognition of text on a video frame can be accomplished by the use of Optical Character Recognition (OCR) tools, an approach investigated in [35]. We analyzed a suite of OCR tools for their effectiveness in recognizing text in video frames. The following tools selected for a comprehensive evaluation: *GOOCR*, an open source

program available under the GNU Public License, *Tesseract* developed at Hewlett Packard Labs and now managed and improved by Google, and MODI (Microsoft Office Document Imaging) toolset. It is discovered that OCR tools generally have limited effectiveness at recognizing text in the presence of 1) certain combinations of text and background colors and shades, 2) text mingled with colorful shapes, and 3) small and exotic fonts. An example set of video frames that were challenging for OCR tools are shown in Figure 5.1.

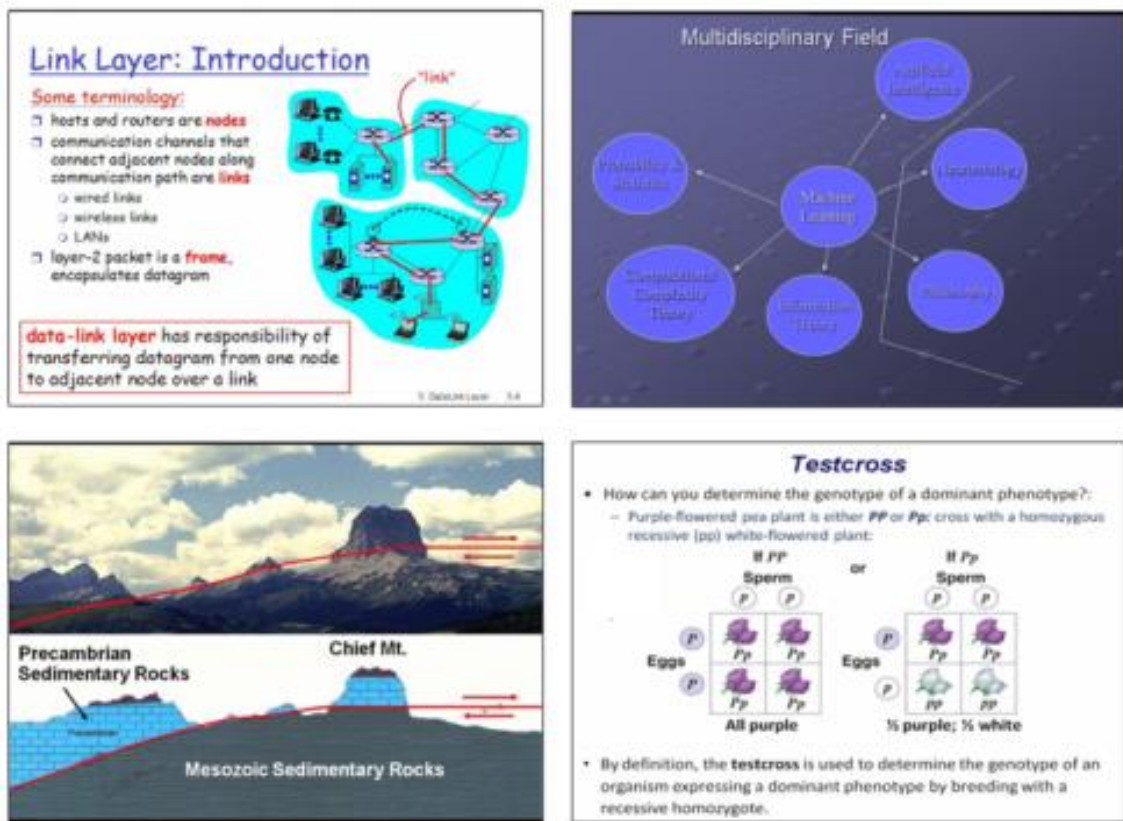


Figure 5.1: Example of ICS video frames which is a challenge for OCR

5.2 Image Enhancement

To increase the detection efficiency of text in video frames, we investigated the use of several simple image processing techniques for image enhancement (IE) prior to the application of OCR tools. IE operations that were effective in enhancing text recognition included *segmentation* of text followed by enlargement with interpolation, and *color inversion*.

5.2.1 Segmentation

Segmentation of text involves steps necessary to define and extract the text regions in an image as shown in Figure 5.3.

Binarization: Text segmentation starts by converting the color image to a binary black and white image by using Simple Image Statistics based thresholding. Threshold is calculated as the sum of weighted pixel values divided by the sum of weights. Binarized image is shown in Figure 5.3b.

Dilation: After binarization of the image, we use dilation, which is removing object holes of too-small a size. This is a morphological operations to connect the characters. We used the following linear structuring element: $[0,0,0;1,1,1;0,0,0]$. The dilation operating in effect allows for expansion of separate objects, or merging of objects in close proximity. We use it for joining the characters and creating groups for identifying a text region in the image. The structuring element is a horizontal window so that the characters tend to merge in the right and left direction in the

image. A single dilation operation is not enough to merge all characters, so the operation is performed 8 times, which we found to be sufficient to join the characters in most ICS video images. An example output is shown in Figure 5.2.










Original Image	
Dilation #1	
Dilation #2	
Dilation #3	
Dilation #4	
Dilation #5	
Dilation #6	
Dilation #7	
Dilation #8	

Figure 5.2: Dilation effect on an image: Dilation joins the small objects (characters) and fills the small holes; text is converted to square objects

Edge Detection: We grouped every small object, such as characters and markings, into a single object by the dilation process. Nonetheless, there can still be incomplete borders after dilation, which may lead to incorrect segmentation. There are several edge detection algorithms and we choose the Sobel operator, which is one

of the most commonly used edge detectors in image processing [44].

Blob Extraction: Blob extraction is used to extract standalone objects in the image and, for our purpose, we want to extract only the regions that have text in them. We count and extract standalone objects in the image using a connected-component labeling algorithm [16]. We use blob extraction to detect the location of text in the dilated image. Blob extraction step is shown in Figure 5.3d. In the extracted blob, one would expect more blobs; however, they were filtered using the following two criteria. If a blob contains other blobs, or if the blob-width / blob-height < 1.5 , it is not extracted. (The text we want to detect is at least two characters long; since we dilated text to the right and left, in all cases the width will be more than the height.). In Figure 5.3d, the man’s body is not extracted because of the threshold on height to width ratio. In addition, very small size blobs are also not included in the extracted regions.

Resizing: Enlargement with interpolation is implemented for the segmented blocks. By this operation, small size text is enlarged to become visible to OCR engines. Resizing is illustrated in Figure 5.3f.

5.2.2 Color Inversion

Color inversion is done by altering the RGB values of images, aimed at increasing the contrast between the text and background. In image file formats such as BMP, JPEG, TGA, or TIFF, that are common in 24-bit RGB representations, color value for each pixel is encoded using 24 bits per pixel, where three 8-bit unsigned integers (0

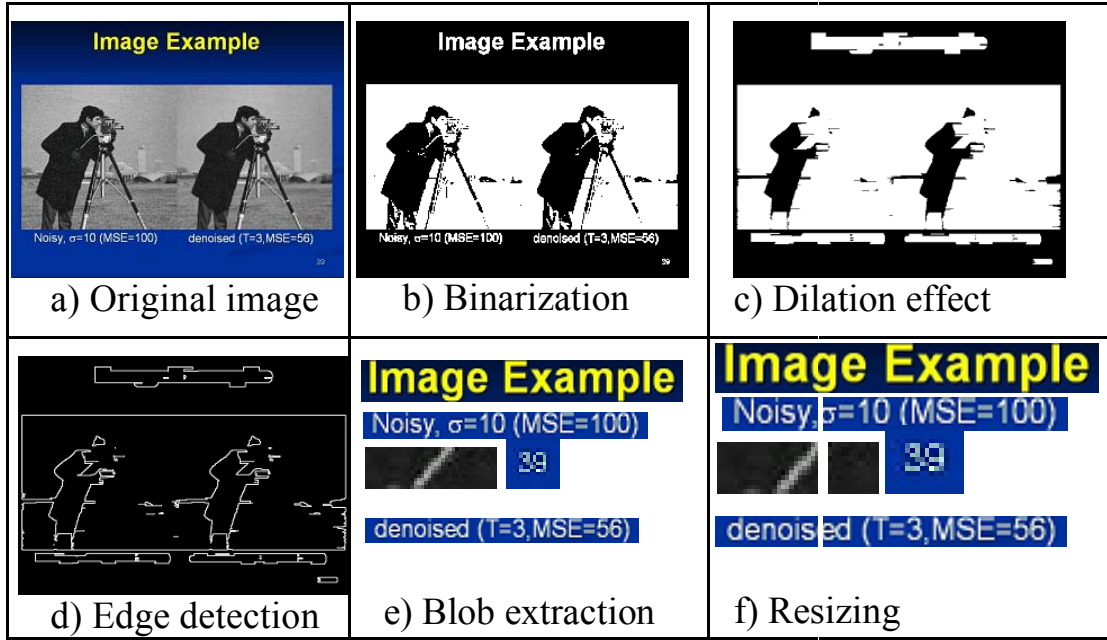


Figure 5.3: Segmentation and enlargement of text

through 255) represent the intensities of red, green, and blue, respectively. Inverting colors is basically altering the RGB values. When we invert an image in a classical way, we take the inverse RGB values. For example, the inverse of the color (1,0,100) is $(255-1, 255-0, 255-100) = (254, 255, 155)$. In our approach, we expand this technique from 1 to 7 inversions shown in Figure 5.4, where R' is referring to $255-R$ value. OCR engines give different results for inverted images. In this example, the image with the 3rd inversion is more clear than the first one. But this will change in different images that have various color combinations.

Image enhancement procedures often lead to new text being recognized, but can also prevent the recognition of other text. Hence OCR engines are applied to the original images as well as the enhanced images and the union of the results is taken.

<u>Original Image</u> <u>R/ G/ B</u>	
<u>Inversion 1</u> <u>R'/G /B</u>	
<u>Inversion 2</u> <u>R /G'/B</u>	
<u>Inversion 3</u> <u>R/G/B'</u>	
<u>Inversion 4</u> <u>R'/G'/B</u>	
<u>Inversion 5</u> <u>R/G'/B'</u>	
<u>Inversion 6</u> <u>R'/G/B'</u>	
<u>Inversion 7</u> <u>R'/G'/B'</u>	

Figure 5.4: Inversion example: Original image and color inverted images. It is an open question that which image is more readable and which image will have better OCR results.

5.3 Evaluation

To test the OCR Tools and the impact of IE procedures, we evaluated 1387 different images that were selected by the indexer from 20 diverse videos. Images in these videos contain 20,007 unique words, 27,201 total words (of more than 1 character length) for a total of 144,613 characters. Search accuracy is defined as the number of detected unique words divided by the total number of actual unique words. Experimental results, presented in Figure 5.5, show that the search accuracy of three distinct OCR engines, Tesseract, GOCR and MODI, improved, with an increase of 9% on average, with IE transformations-union of text segmentation and color inversion. Segmentation and inversion both increased the accuracy, but inversion is slightly more effective than segmentation. The maximum accuracy obtained by applying all OCR engines with image enhancements was 97.1%. Alternately stated, the miss rate was 8.9% for the best single OCR engine, 5.2% for all OCR tools combined, and 2.9% for all OCR engines combined with image enhancement.

Image enhancement provided this accuracy improvement, but increased the processing time significantly, partly because OCR engines have to be applied on the original and the enhanced images. Nonetheless, the processing time remains modest for a typical video. On average it is in the range of 2-3 minutes for an hour long video on a typical desktop. Image enhancement also doubled the false positives detected by OCR engines, i.e., more words were detected that were not actually present in the video. This often happens when an OCR engine misses a character in a word, leading to false identification of a different word. Since the main aim of the text

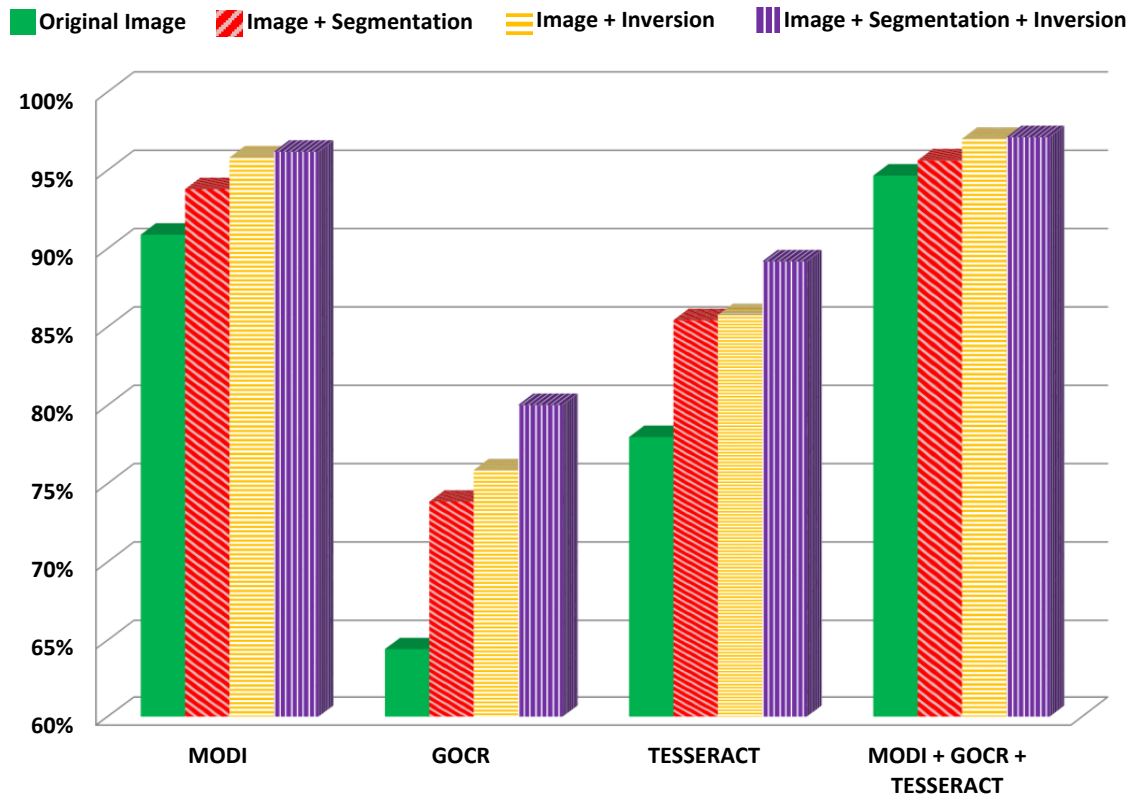


Figure 5.5: Search accuracy rate of OCR tools

recognition is to let the user find words of interest, the extra words resulting from false positives are unlikely to diminish the functionality in a significant way.

Chapter 6

Ground Truth Creation and Accuracy Metrics

Text-based video indexing algorithms were evaluated for their indexing accuracy. The evaluations helped to ascertain the strengths and weakness of the base algorithm, which prompted further enhancements. The text-based indexing was tested on twenty-five different videos, of which ten are video recordings from lectures conducted at University of Houston (hence UH) and fifteen videos are from Coursera. The following Table 6.1 provides data on the course and number of lectures for each course. The consent of the instructor to provide the ground truth data is required for the evaluation of another factor in the selection of videos.

Table 6.1: List of Department and Courses used for indexing experiment

Type	Department	Course	Number of Lecture
UH	GEO	Physical Geology	1
UH	COSC	Digital Image processing	3
UH	COSC	Computer Architecture	2
UH	COSC	Computer Network	2
UH	COSC	Computer Vision	2
Coursera	COSC	Compilers	3
Coursera	COSC	Cryptography	3
Coursera	COSC	Machine Learning	2
Coursera	COSC	Probabilistic Graphical Models	2
Coursera	COSC	Data Science	3
Coursera	COSC	Natural Language Processing	2
Total			25

6.1 Ground Truth and Rating of Transition Points

The ground truth defines the actual index points among all the transition points for a video. However, determining the ground truth is not easy. Some transition points are difficult to differentiate as a start of a topic. The ground truth depends on the perception of the viewer and can be different for each individual. Even the experts who created the video may have difficulty in identifying the index points. For this reasons, every transition point in a lecture video is rated from 0 to 3. The rating indicate whether the given transition point is a good candidate for index point or not and is based on the following conditions:

- **Definitely an index point (rating of 3):** If a given transition point is found to be definitely an index point or start of a new topic, it is rated as 3.
- **Probably an index point (rating of 2):** Certain transition points may not

appear to be a definite index point, i.e., start of a new topic. The reasons could be such as the introduction of a sub-topic that is part of the main topic, an example given for clarity and so on. It is difficult to differentiate the given transition point as a definite index point. A rating of two indicates a high probability that the current transition point is an index point.

- **Probably not an index point (rating of 1):** Likewise, a rating of one indicates that a transition point is probably not an index point.
- **Definitely not an index point (rating of 0):** If the transition point is definitely not an index point, a rating of zero is given.

The ratings for UH collected from instructors from the slide handouts or entered via the interface we have designed as shown in Figure 6.1. The grader can see the frames and frame texts based on the relationship with its previous and after frames he can give a score between 0 and 3. The indexing algorithm output for each video was evaluated against its ground truth. As explained previously, a typical lecture video has 20 to 100 transition points. The index points are a sub-set of these transition points. The ground truth of each video was marked manually, i.e., a transition point is marked as an index point or not, depending on its suitability. The respective instructor who presented the lecture provided the ground truth data for the videos from the University of Houston. Each individual video is a single, continuous recording of a single classroom lecture. On the other hand, video lectures presented in Coursera are separate recordings of the subtopics that constitute the whole subject or topic. The Coursera web interface is shown in Figure 6.2, allows

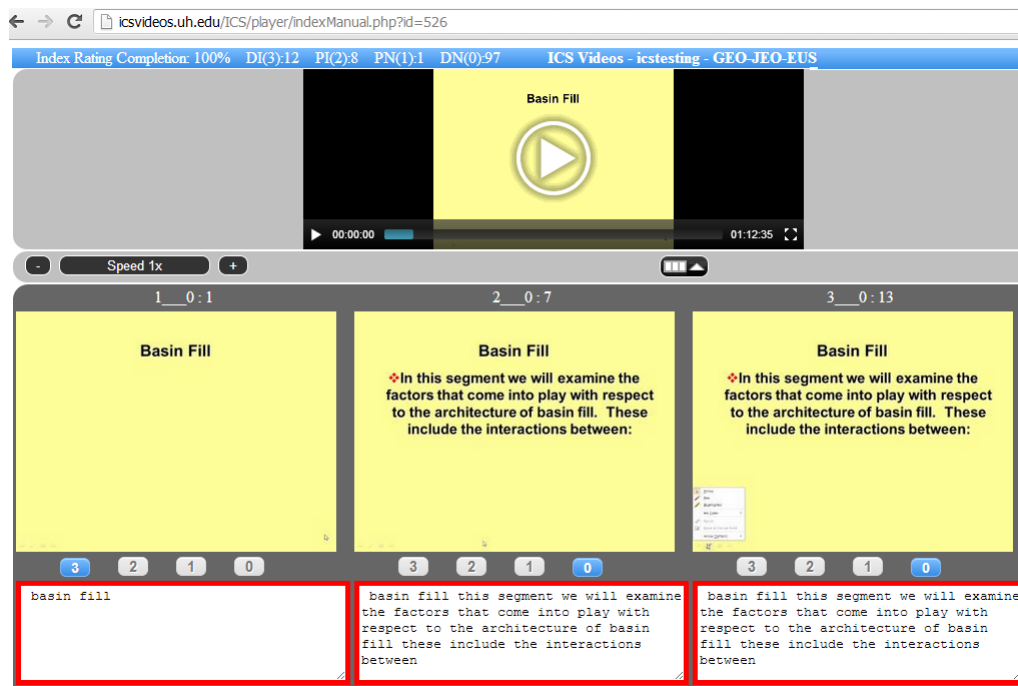


Figure 6.1: Interface for ground truth index point creation: 3-Definitely index points, 2-Probably index point, 1-Probably not an index point, 0-Definitely not an index point

access to these individual segments or sections. These individual segments merge to form a single video and forms the input to ICS framework for indexing evaluation. The merging is a manual process. The ground truth data for the Coursera videos are the individual sections that form the complete topic. In other words, the transition points that represent or correspond to the start of these individual segments are marked as the ground truth index points. Table 6.2 tabulate the ground truth data for University of Houston and Coursera. The table lists the number of index ratings, total transition points, and duration of 25 videos.

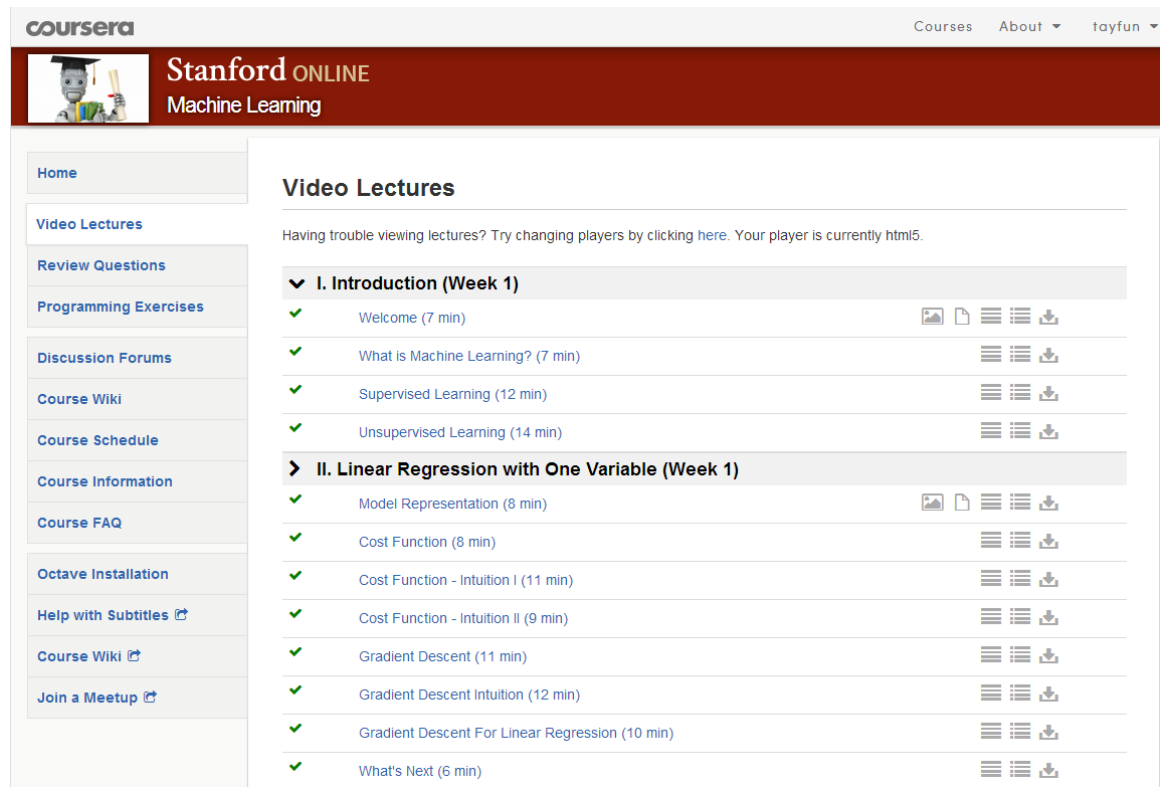


Figure 6.2: Coursera interface to access the videos: each video is divided into segments

Table 6.2: Ground Truth for 25 Lecture Videos

Video ID	# of Definitely Not Index Points(0)	# of Probably Not Index Points(1)	# of Probably Index Points(2)	# of Definitely Index Points(3)	Total # of Transition Points	Total Video Duration in Minutes
1	22	0	0	8	30	48
2	78	0	0	4	82	54
3	36	12	10	5	63	77
4	53	25	12	9	99	77
5	23	1	5	2	31	85
6	43	15	9	3	70	83
7	24	4	4	7	39	72
8	22	0	4	1	27	76
9	2	12	3	2	19	72
10	99	0	2	6	107	82
11	41	0	0	4	45	46
12	46	0	0	5	51	80
13	72	0	0	8	80	81
14	99	0	0	5	104	99
15	49	0	0	5	54	60
16	78	0	0	6	84	92
17	110	0	0	3	113	36
18	112	0	0	5	117	81
19	59	0	0	6	65	63
20	71	0	0	7	78	75
21	55	0	0	6	61	67
22	79	0	0	8	87	45
23	55	0	0	9	64	67
24	27	0	0	4	31	24
25	24	0	0	3	27	25
Total	1379	69	49	131	1628	

The video duration, the sub-topics, or the slides that constitutes the lecture can be different for each selected video, as evident from the ground truth data presented in tables Table 6.2 and Table 6.3. Therefore, the index points for individual videos also can vary, which makes the evaluation challenging. The unique nature of the video indexing problem motivated the development of a custom scoring metric for the evaluations.

6.2 Indexing Score Calculation

The output of the indexing algorithm as well as the ground truth forms the basis of Indexing Score. However, different metrics are needed for comparing various outputs of the algorithms.

6.2.1 2-Point Metric

Accuracy is used as a statistical measure of how well a binary classification test correctly identifies. It is the most common evaluation metric including the true positives(tp), false positives (fp), true negatives (tn) and false negatives (fn) as shown in Table 6.3. The accuracy is calculated by the formula below:

$$\text{Accuracy} = (\text{tp} + \text{tn}) / (\text{tp} + \text{tn} + \text{fp} + \text{fn})$$

This metric is called as “2-Point Metric” to differentiate from others since both the input and ground truth is binary, IP and Not IP. In this 2-point metric system, some of the other measurements are listed in below :

Precision= $tp/(tp+fp)$

True positive rate, sensitivity, recall= $tp/(tp+fn)$

True negative rate= $tn/(tn+fp)$

Table 6.3: 2-Point scale accuracy matrix

		Ground Truth	
		IP	Not IP
Test Outcome	IP	(true positive)	(false positive)
	Not IP	(false negative)	(true negative)

6.2.2 4-Point Metric

The Indexing Score for each transition point of the lecture video satisfies the conditions listed below.

- If the given transition point is found to be an index by the indexing algorithm and is marked as definitely an index point in the ground truth (rating of 3), an indexing score of 2 is given to that transition point. Conversely, if the indexing algorithm marks the transition point as not an index point, it is scored as 2.
- If the given transition point is found to be an index by the indexing algorithm and is marked as probably an index point in the ground truth (rating of 2), an

indexing score of 1 is given to that transition point. Conversely, if the indexing algorithm marks the transition point not an index point, it is scored as 1.

- If the given transition point is found to be not an index by the indexing algorithm and is marked as probably not an index point in the ground truth (rating of 1), an indexing score of +1 is given to that transition point. Conversely, if the indexing algorithm marks the transition point as an index point, it is scored -1.
- If the given transition point is found to be not an index by the indexing algorithm and is marked as definitely not an index point in the ground truth (rating of 0), an indexing score of 2 is given to that transition point. Conversely, if the indexing algorithm marks the transition point as an index point, it is scored 1.

Table 6.4 provides a summary of the scoring conditions. The sum of scores of each transition points gives the total indexing score. The total indexing Score is calculated using the following formula:

$$\text{Indexing score} = \sum_{k=1}^n (\text{Transition point indexing score}).$$

Where n is the total number of transition points in the video.

The theoretical maximum score for a video is the total sum of theoretical scores of the transition points in the video and is based on the following conditions:

- If the given transition point is rated as 3 or 0, a theoretical score of +2 is given to that transition point.

Table 6.4: Possible indexing scores for a transition point with 4-Point scale

		Ground Truths			
		Definitely Not IP	Probably Not IP	Probably IP	Definitely IP
Algorithm Output	0 (Not IP)	(+2)	(+1)	(-1)	(-2)
	1 (IP)	(-2)	(-1)	(+1)	(+2)

- If the given transition point is rated as 2 or 1, a theoretical score of +1 is given to that transition point.

The reason behind the scoring scheme is that, the output of an ideal indexing algorithm will always matches the ground truth, i.e., the definite and probable index points will always be marked as index points by the algorithm and vice versa.

The following formula gives the Theoretical maximum score. theoretical maximum

$$\text{score} = \sum_{k=1}^n (\text{transition point theoretical score})$$

Where n is the total number of transition points in the video. A Scoring Metric makes the evaluation as well as the relative comparison of the indexing output easier. Based on the output of the indexing algorithm, the Scoring Metric gives the accuracy score for a video. The Indexing Accuracy forms the criteria for evaluation or comparison between the indexing algorithms. Two phases are involved in the Indexing Accuracy calculation: 1) the calculation of the Theoretical Maximum score for the video, 2) the calculation of the indexing score. Rating every transition

point in the video forms the basis of calculation the theoretical maximum score for a video. The instructor provides the rating for the University of Houston videos. The topic segmentation provided in the Coursera web interface forms the basis of Coursera video ratings. A Theoretical maximum score is the maximum possible indexing score attainable for a given video. The Ground Truth rating provides the data for its calculation. The output of the indexing algorithm forms the basis of indexing score calculation for the given video. The following formula defines Indexing Accuracy.

$$\text{indexing accuracy} = (\text{indexing score}) / (\text{theoretical maximum score})$$

6.2.3 Sorting Metric

Sorting metric is used to compare a sorted list to another sorted list. If the outputs of the algorithm is an ordered list and the ground truth can be sorted, this metric is used. Sorting Metric calculates the total distance of each pair in different orderings. Steps of scoring metric is depicted in Figure 6.2.3. In this example, algorithm1 ordering is BACED and algorithm2 ordering is ACEBD, while the ground truth order is ABCDE. Algorithm1 is compared to algorithm2 based on the ordering of ground truth to clarify which ordering is more close to ground truth? This is calculated as in Figure 6.2.3. For algorithm1, distance of first element B to distance of B in ground truth is 1. Distance of A is also 1. Order of C in algorithm1 and ground truth is same, 3, as a result the distance of C is 0. The rest is calculated in this way. As a result the sum of distances of algorithm1 is 4 (1+1+0+1+1) and sum of distances of algorithm2 is 6 (0+1+2+2+1). This shows that algorithm1 is more close to ground truth than algorithm2 since its ordering distance to ground truth order is less.

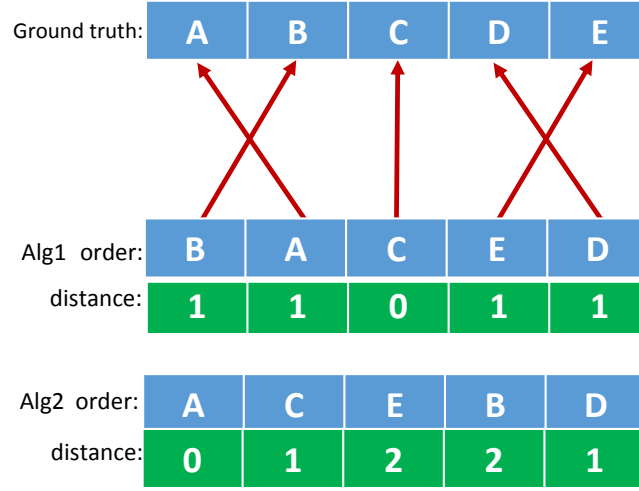


Figure 6.3: Scoring algorithms with sorting metric

The reverse order of ground truth is the maximum distance that an algorithm can have. So if the distance score is divided to maximum distance, a normalized distance score can be calculated. In this case, the sorting score will be the subtraction of this ratio from 1:

$$\text{Sorting Score} = 1 - \text{distance} / \text{maximum distance}.$$

Sorting metric is different from Kendall's Tau distance([27]). It calculates the distance between of each pair(one in the algorithm, other in ground truth) only once in the global orderings rather than calculating of each consecutive pairs and getting sum of it as done in Kendall's Tau. We are not interested in relations of consecutive pairs in the output of an algorithm but interested in the position of the item in an algorithm compare to ground truth ordering. Figure 6.2.3 depicts how to apply the sorting metric to ground truth to compare algorithms. Ground truth has 5 transition points: T1, T2, T3, T4 and T5 and their index ratings which were given

by the instructors are 3,0,1,2,0 consecutively. First step is to sort these transition points in the ground truth based on these index ratings. When it is sorted the order of ground truth becomes T1-T4-T3-(T2 and T5)- (T2 and T5). As can be observed that T2 and T5 is repeated because they both have same index ratings, in this case it is 0, which means they can be used interchangeably. In fact, two transition points having the same index ratings cannot be sorted they should be treated equally. Alg1 and Alg2 have very similar orderings the first three transition points are in the same order. Whereas, the order of last two item T2 and T5 is different. Since the T2 and T5 had same index rating in the ground truth, the distance score is same for these two algorithms by our sorting metric. If we try to score these two algorithms with Kendall's Tau distance metric by using this ground truth, it will treat T2 and T5 differently and the score of these two algorithm will be different. It will prefer on algorithm two other which is a case we cannot accept.

Ground Truth	T1	T2	T3	T4	T5
	3	0	1	2	0
Ground Truth Sorted	3	2	1	0	0
	T1	T4	T3	T2 T5	T2 T5
Alg1 order	T1	T4	T3	T2	T5
distance	0	0	0	0	0
Alg2 order	T1	T4	T3	T5	T2
distance	0	0	0	0	0

Figure 6.4: Applying sorting metric to score algorithms with ground truth

Chapter 7

Text-based Indexing

This section discusses the basic text-based indexing algorithm and the various enhancements.

7.1 Fixed Grouping Text-based Indexing Algorithm

The Fixed Grouping is the basic text-based indexing algorithm. Other text-based algorithms are variations or enhancements of the Fixed Grouping algorithm. It closely follows the previously discussed Uniform algorithm. However, the text similarity of the segments decides which neighbor to merge the smallest segment. The detailed explanation of text similarity calculation is provided in Chapter 6. This algorithm compares the text of the smallest segment with a group of segments on its right as well as on the left side as shown in Figure 7.1. Empirically selected value of Grouping Duration determines the number of segments for grouping so that the combined

duration of the group should not exceed the Grouping Duration. The grouped segments are considered as a single segment. The addition of text in the individual segments forms the group. For the similarity comparison against a group, the text of a given segment is compared with the combined text of the group. The algorithm is explained as follows.

Data: A list of transition points ;

Required number of index points (N) ;

Grouping duration in seconds

Result: N index points which is a subset of transition points

repeat

 Select transition point with smallest duration;

if *the similarity is more towards right group* **then**

 merge right;

else

 merge left

end

until *Number of transition points == Required number of index points;*

Algorithm 2: Fixed grouping text-based indexing algorithm

A pictorial example of the algorithm is provided in Figure 7.1. In this example, the similarity of the smallest segment K is compared with the left as well as the right group and merged with the most suitable neighbor depending on the similarity value.

Grouping of several transition points involve combining all the transition points into a single segment. When two segments are added, the respective term frequency

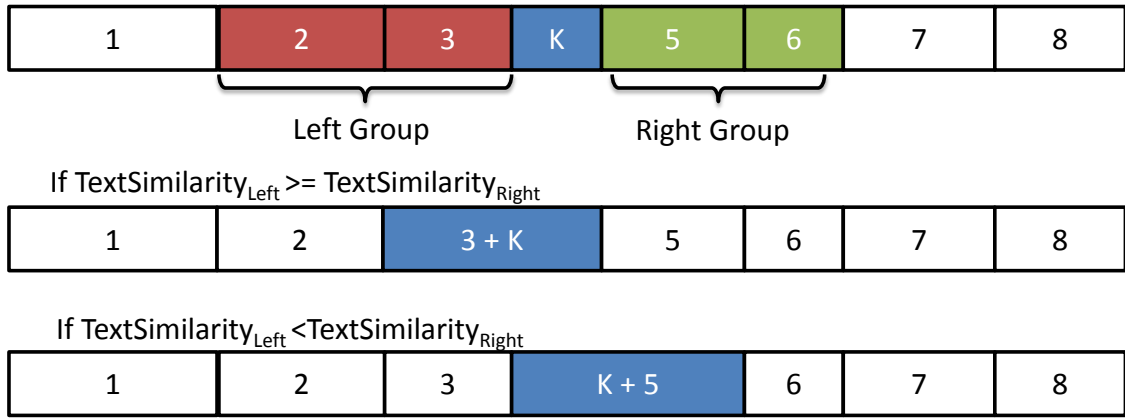


Figure 7.1: Fixed grouping algorithm example: Shortest segment compared to left and right group of neighbor segments and how it is merged based on similarity

vectors for the segments are added together resulting in a new term frequency vector. For the similarity comparison to the left side of a given segment, segments on the immediate left side of the given segment are grouped together and vice versa for the similarity comparison to the right side. The combined duration of the group of segments is not to exceed an empirically determined value of Grouping Duration. The similarity of any given transition point is not with its immediate neighbor, but across several segments over a period. A topic transition takes place over several segments and typically, the transition is not abrupt. In addition to the above, some neighboring segment or in between segment may contain very low text. This may result in incorrect similarity comparison. The text of any single segment may not sufficiently represent the topic entirely. The grouping also aims to reduce the segregation of index points that are part of the same topic, caused by transition points having very low or no text content. The following Figure 7.2 shows an example of why the grouping is necessary.

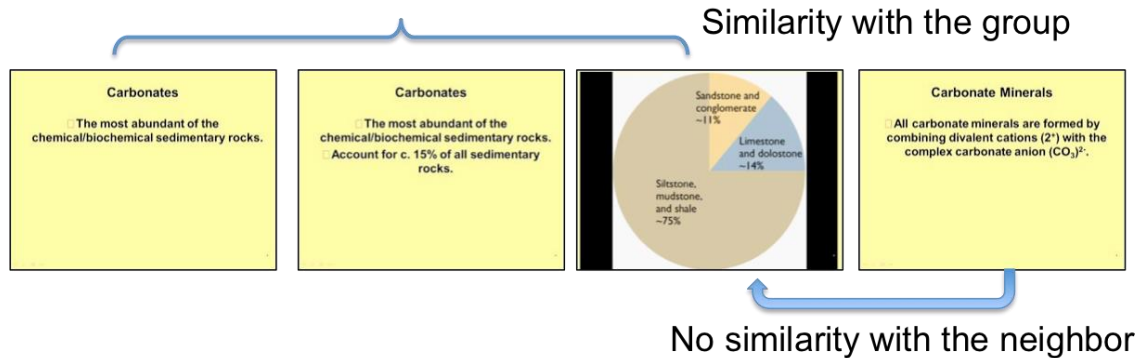


Figure 7.2: An example of reason for fixed grouping of segments

The segments or frames that fall inside a fixed window of time duration are grouped together in fixed grouping algorithm. The time duration is empirically determined based on the tests conducted on the sample video set. However one drawback of using a fixed window for grouping is that, a relevant segment could be ignored due to the time duration limitations because the ignored segment falls outside the fixed grouping duration limit. This requires an enhancement of considering all the segments until the end or beginning of the video.

7.2 Linear-weighted Text-based Indexing Algorithm

The Linear-weighted text-based indexing algorithm is an enhanced version of the previously discussed Fixed-grouping text-based algorithm. This algorithm aims to improve on the limitation of the Fixed-grouping algorithm. The Linear-weighted algorithm eliminates the grouping of the transition points; instead, all the segments in the video are considered. The closer and larger segments are more important when considering all the segments. Therefore, this algorithm introduces weighting

of segments for providing appropriate weights or varying impact on the similarity calculation across the segments. This algorithm adds an additional step to the Fixed-grouping algorithm which involves computing the linear weighted similarities at each transition point. The next step is the transition point selection and merging, exactly as the Fixed Duration algorithm. The detailed algorithm is provided in the following section.

Data: A list of transition points ;

Required number of index points (N)

Result: N index points which is a subset of transition points

Compute linear weighted similarities ($WS_1, WS_2, WS_n - 1$) at each transition points ($T_1, T_2, T_n - 1$) considering segments until the end;

repeat

 Select transition point with smallest duration;

if *the weighted similarity is more towards right* **then**

 merge right;

else

 merge left

end

until *Number of transition points == Required number of index points;*

Algorithm 3: Linear-weighted text-based indexing algorithm

The Linear Weighting algorithm considers all the segments to the end for the similarity calculation as an enhancement to the basic fixed grouping. The entire segments on the left side of the current segment form a left group. Grouping together all the transition points on the right side of the current segment forms a right

group. Considering all the segments ensures that no segments are ignored during the similarity comparison and is supposedly more accurate. Because by comparing with the complete set of segments to the left or right provides more information, a better decision can be made when regarding which direction to merge. However, considering all the segments can cause other problems. A segment far away can cause equal influence on the merging decision as a segment that is nearby leading to the possibility of merging to the wrong side. Clearly, the closer segments should have more weightage towards the merging decision. The transition point creation phase can split a segment of long duration into smaller segments. In this scenario, the individual smaller segments contain the same terms set as the original larger segment. These smaller segments when grouped together lead to an increase in the term frequency count, causing an increased undue influence on the merging decision. Ideally, there should not be such an excessive influence. Usage of a proper weighting scheme address these issues, which forms an essential part of the enhancements and is discussed in the following paragraphs.

For the similarity calculation, each transition point is given a weight based on the duration of the segment as well as the time or distance the frame is away from the current segment under consideration. The Time-based weight reduces linearly. In linear weighting, the weight of a segment is determined linearly based on how far in time the segment is away from the current segment. Segments that are closer in time carry more weight. The weight reduction is in a linear manner and inversely proportional to the time between the segments. Linear weighting contains two components, duration based and time based weight. transition points with longer duration need

more weight than transition points with shorter duration. This ensures that the effect of similarity calculation of a single long segment is same even if the same long transition point splits into multiple shorter pieces when grouped together. Without the Duration-based weight, these short segments, when grouped together, result in a higher similarity value than that of a single segment of longer duration. However, the smaller segments are formed because of splitting the single longer segment into several transition points of shorter duration. This is because, the shorter segments still have the same word frequency vector as the longer segment. Duration-based weight component, W_d , is given by the following formula:

$$W_d = \text{transition point duration} / \text{Total video duration}$$

When several transition points are grouped together for similarity comparison, segments that are farther away from the current segment should contribute lesser to the segment that are closer to the current segment. Time-based weight is linear interpolated based on the time difference between the transition points under consideration and is given by the following formula. Time-based weight $W_t = 1 - (\text{Time difference between frames} / \text{Total video duration})$ Figure 7.3 indicates the linear weight reduction of the segments across the entire video.

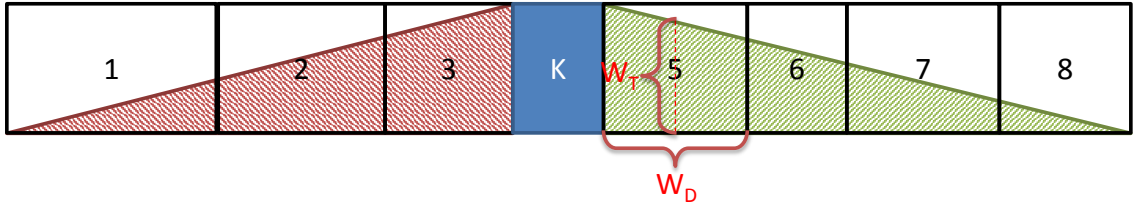


Figure 7.3: Linear weight reduction of the transition points

Total weight is the product of Duration-based weight and Time-based weight and is given by the formula:

$$W = W_d * W_t.$$

The major disadvantage of linear weight reduction is that the farther segments are still significantly influencing the similarity value. The solution is to reduce the weight at a faster rate, such as in a non-linear fashion.

7.3 Non-linear Weighted Text-based Indexing Algorithm

This algorithm is an enhancement to the Linear-weighted algorithm. Algorithm works the same as the Linear-weighted algorithm, except that the weighting changes to non-linear weight reduction. The detailed algorithm is provided in the following section.

Data: A list of transition points;

Required number of index points (N)

Result: N index points which is a subset of transition points

Compute non-linear weighted similarities ($WS_1, WS_2, WS_n - 1$) at each transition points ($T_1, T_2, T_n - 1$) considering segments until the end;

repeat

 Select transition point with smallest duration;

if *the weighted similarity is more towards right* **then**

 merge right;

else

 merge left

end

until *Number of transition points == Required number of index points;*

Algorithm 4: Non-linear weighted text-based indexing algorithm

In a non-linear based weighting reduction, the time-based component of the total weight reduction is non-linear instead of linear reduction. This ensures a faster rate of reduction than the linear reduction. An arbitrary variable called half-life determines the rate of decay. Half-time is the time where the weight becomes half and is heuristically determined. An exponential weight decay function is used in this weight calculation and is given by the following formula:

Where the rate of decay is calculated using the following equation:

The following Figure 7.4 represents the typical weight decay for various values of half-life.

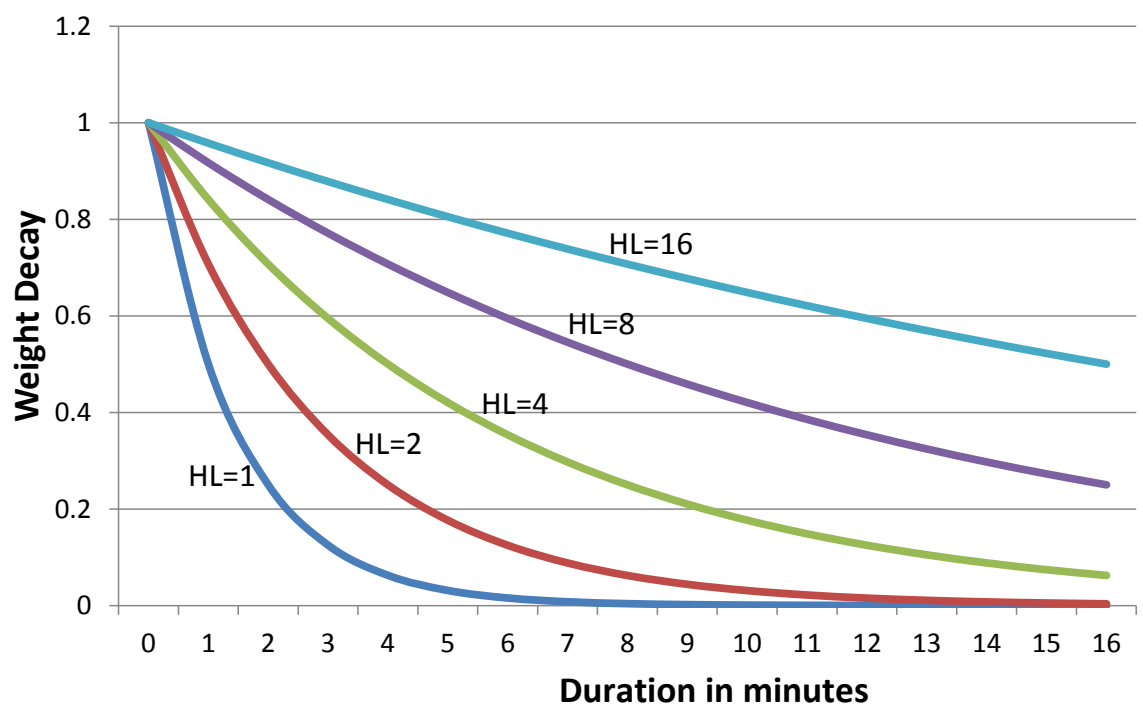


Figure 7.4: Weight decay for various half-life

The total non-linear weight is the area under the weight decay curve and is given by the following formula. The area under the curve is sampled at a sampling duration of 20 seconds, and the sum of the areas of all the samples gives the total non-linear weight.

where WSD is the sampling duration weight and $FNL(t)$ is the non-linear function.

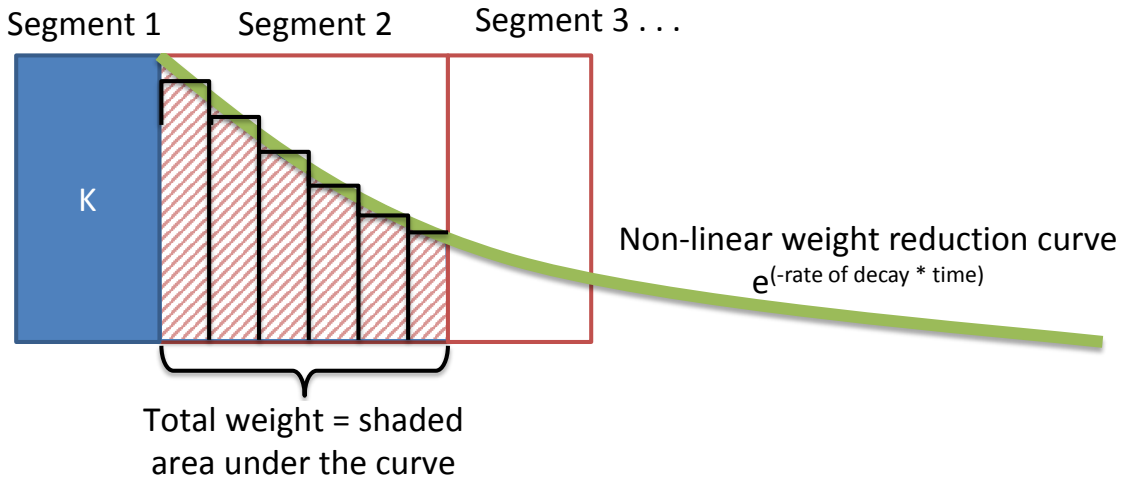


Figure 7.5: Non-linear weight reduction

7.4 Boundary-based Text-based Indexing Algorithm

Boundary-based text-based indexing algorithm is a new algorithm where the selection of index points depends on the transition points where the similarity is the least. Similar to the Non-linear Weighted algorithm, this algorithm computes the weighted similarities at each transition points. transition points with the least weighted similarity are selected as index points. The algorithm is detailed as follows.

Data: A list of transition points;

Required number of index points (N)

Result: N index points which is a subset of transition points

1. Compute non-linear weighted similarities ($WS_1, WS_2, WS_n - 1$) at each transition points ($T_1, T_2, T_n - 1$) considering segments until the end;
2. Sort the transition points in ascending order based on weighted similarity;
3. Declare the first N transition points with the lowest similarity as index points;

Algorithm 5: Boundary-based text-based indexing algorithm

Following Figure 10 provides an example of the selection of boundaries based on least similarities. transition points 1, 4, and 6 are selected, as index points since the corresponding weighted similarities are the lowest. At each boundary, the current topic transitions into a new topic. The algorithm is based on the assumption that the text similarities between different topics are lower than the text similarity of segments within the same topic.

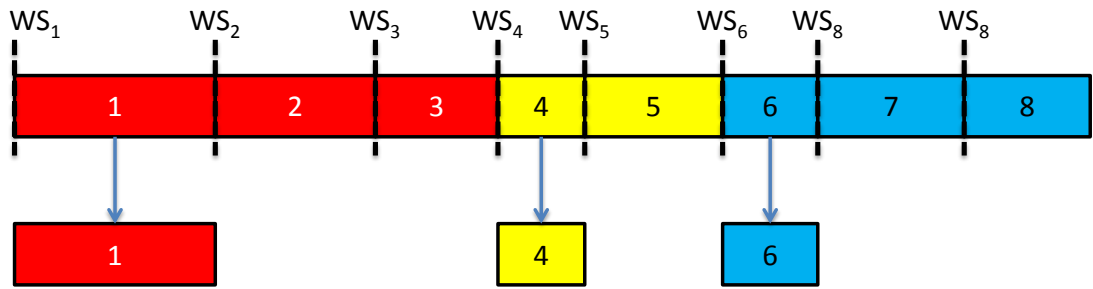


Figure 7.6: Boundary-based index point selection

The Boundary-based algorithm eliminates the selection of transition points based on the smallest duration. Smaller duration segments could possibly be the start of

an index point. In the previous algorithms, the chances of merging such transition points are very high. Therefore, the Boundary-based selection has the potential to select such smaller transition points as index points. A disadvantage of this approach is that, this selection could result in very close or far spaced index points.

7.5 Term Frequency-Inverse Document Frequency (TF-IDF) Optimization

This is an optimization applied to the previously discussed algorithms. The TF-IDF optimization compensate for the effect of common words that appear in many transition points that may not contribute towards the topic similarity. In term frequency-inverse document frequency or TF-IDF weighting [34, 41], a word that is repeated in many transition points are given low weight and a term having a high frequency within a transition point, but not repeated across many segment are given more weight. Several parts of speech like articles, conjunction, preposition, etc. could be used in the construction of the slide and they could be repeated in several transition points. These terms do not contribute to the meaning or differentiation of a topic from another and therefore are considered as noise, thereby masking the influence of the important terms (nouns) that make up the topic. The terms that contribute towards a particular topic are supposed to be concentrated in that topic and generally repeat or appear only in a small number of segments that forms a sub-topic. Therefore it is important that the common terms that are repeating throughout the segments should be given lesser importance than the topic keywords

of terms that contribute towards a particular topic and this can be ensured by TF-IDF weighting scheme.

Term (T) is a word that is present in any frame. A term may not appear in certain frames. Nevertheless, it is present in at least one frame. Term frequency (TF) is the frequency of a term T in a frame F. Term frequency is a scalar value. Inverse document frequency (IDF) indicates whether a term T is common or rare across all transition points and the following formula gives the IDF;

$$\text{idf}(t,f) = \log \frac{|TS|}{1 + |\{f: t \in f\}|}$$

where:

$|TS|$ is the total number of transition segments under consideration or the cardinality of TS

$\{F : T \in F\}$ is the number of segments where the term t appears

Term frequency-inverse document frequency (TF-IDF) assigns a weight to a term T in a frame F given by the formula:

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

The weight of the term reduces logarithmically with the increase in the number of frames in which the term is present.

7.6 Evaluation

Experiment result on the text-based indexing algorithms and uniform indexing algorithm is displayed in Figure 7.7. Desired number of index points are given equal to the number of index point in ground truth. Average accuracy result for all videos with 4-point metric shows that text-based indexing algorithms performs better than uniform indexing. Furthermore, among the text-based indexing algorithms non-linear weight provides highest indexing accuracy.

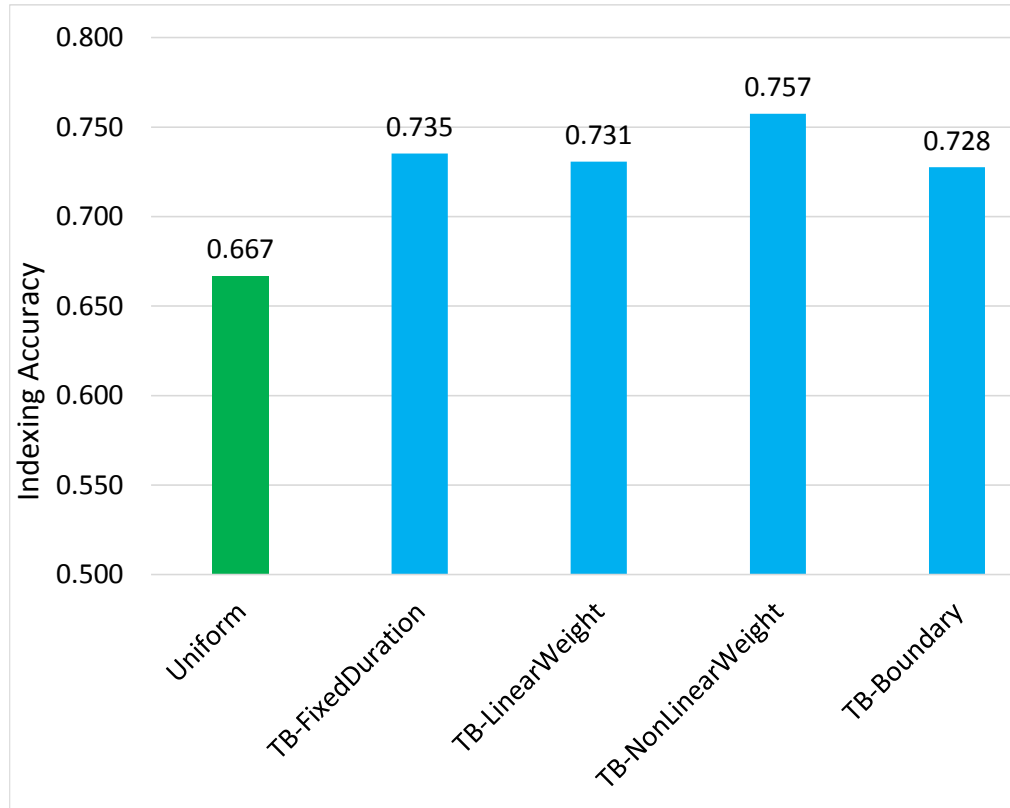


Figure 7.7: Indexing (4-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in the ground truth

Chapter 8

Indexing by Machine Learning

In sum, text-based indexing algorithms consider very limited features:

- duration of *transition point*
- cosine similarity of left group and right group, linear and nonlinear weight

We can create more text-based indexing algorithms by using more features like title of the segments, first time words appear in the video etc.. But the number of features we use will still be limited due to the feasibility for the practical use. So a different approach such as machine learning (ML) can be used. Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions. Machine learning uses the theory of statistics in building mathematical models, and the core task is making inference from a sample[4]. The sample -dataset-

in our indexing approach can be gathered from the ground truth which is created by professor's inputs. The following sections, will examine how the dataset and features are created and details of applying machine learning algorithms for video indexing problem.

8.1 Objective of Machine Learning for Video Indexing

Our main objective is to find out whether we can segment videos into topics by using machine learning algorithms. Ultimately our goal is to have a better video indexing algorithms than text-based indexing algorithm. This objective requires answering other questions like:

- How to use ground Truth Data for ML? What features we can create for dataset?
- What will be the input and the output of ML?
- How can we use the output of machine learning approach when given the limitations of a certain number of index points?
- What features are important for deciding index points?
- How does machine learning indexing performs comparing text-based indexing algorithms?

8.2 Dataset and Creating Feature Vectors

As we discussed in 7, to evaluate the text-based indexing algorithms ground truth was created. Professors defined each transition points as definitely index point, probably index point, probably not index point and definitely not index point. As seen in Figure 8.1, in ground truth data, there is only duration and text for a specific transition point in a video. In the set of features, the word *left* refers to previous transition points, whereas the word *right* refers to following transition points. The words *title5* and *title10* refer to set of 5 words or 10 words having largest font size in slide. Some of the features in dataset created from these two columns are listed in the following.

- **duration:** Duration of transition points in seconds.
- **allWordsCount:** Number of all words, including the repetitions in that transition point.
- **uniqWordsCount:** Number of unique words in that transition point.
- **firstTimeWordsCount:** Number of words that appear in the video for the first time in this transition point. Central idea of this feature is new topics represented by new words. So if a word appears in the video for the first time new topic is being introduced.
- **leftCommonWords1-3:** Number of words in common with the current transition point and the first, second and third previous transition points.

- **leftDuration1-3:** Duration of the first, second and third previous transition points.
- **rightCommonWords1-3:** Number of words in common with the current transition point and the first, second and third following transition points.
- **rightDuration1-3:** Duration of the the first, second and third after transition points.
- **leftCommonWordsAll:** Number of common words with all previous transition points.
- **leftDurationAll:** Total duration of all previous transition points in seconds.
- **rightCommonWordsAll:** Number of common words with all following transition points.
- **rightDurationAll:** Total duration of all following transition points in seconds.
- **left1min to left10min:** Number of common words with the previous slide in 1 to 10 minute distance away.
- **right1min to right10min:** Number of common words with the following slide in 1 to 10 minute distance away.
- **ngram2left1min to ngram2left10min:** Number of common 2-gram sequence with the previous slide in 1 to 10 minute distance away.
- **ngram2right1min to ngram2right10min:** Number of common 2-gram sequence with the following slide in 1 to 10 minute distance away.

- **title5left1min to title5left10min:** Number of common title words(5 words having the largest font sizes) with the previous slide in 1 to 10 minute distance away.
- **title5right1min to title5right10min:** Number of common title words(5 words having the largest font sizes) with the following slide in 1 to 10 minute distance away.
- **title10left1min to title10left10min:** Number of common title words(10 words having the largest font sizes) with the previous slide in 1 to 10 minute distance away.
- **title10right1min to title10right10min:** Number of common title words(10 words having the largest font sizes) with the following slide in 1 to 10 minute distance away.
- **ngram2Title5left1min to ngram2Title5left10min:** Number of common 2-gram sequence in title5 words(5 words having the largest font sizes) with the previous slide in 1 to 10 minute distance away.
- **ngram2Title5right1min to ngram2Title5right10min:** Number of common 2-gram sequence in title5 words(5 words having the largest font sizes) with the following slide in 1 to 10 minute distance away.

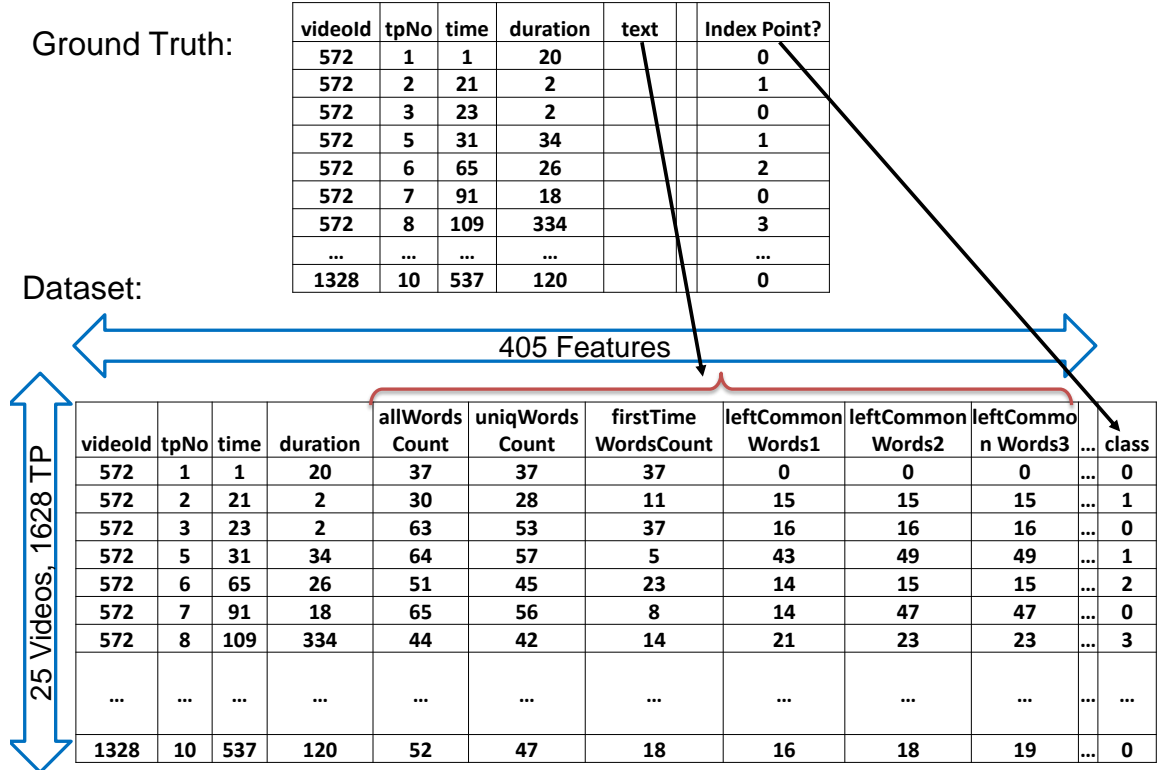


Figure 8.1: Creating dataset for machine learning from ground-truth table

8.3 Handling 4-Level Input and 2-Level Output

In an ideal video indexing algorithm, the output is 2-level: index point or not index point. All the text-based indexing algorithms explained in previous chapter works in the same way. But as we know the ground truth is 4-level, because of that dataset for machine learning is also 4-level as shown in Figure 8.1. In machine learning if the input is 4-level the output will also be 4-level. In machine learning, for 4-level input and 2-level output classification design some adjustments needs to be done as depicted in Figure 8.2.

Figure 8.3 shows that there are 3 possible ways to handle 4-level input and 2-level

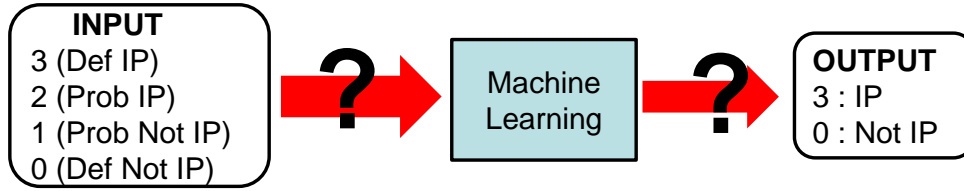


Figure 8.2: 4-Level input and 2-level output challenge in machine learning

output in machine learning approach.

- **Convert Binary After Machine Learning:** In the first approach 4-level input can be processed by machine learning which will create 4-level output. 4 level output (3,2,1,0) can be converted to binary(3-2:IP, 0-1:Not IP) as shown Figure 8.3a.
- **Convert Binary Before Machine Learning:** In this approach 4 level input (3,2,1,0) can be converted to binary(3-2:IP, 0-1:Not IP) before processing with machine learning so that output will be 2 level, Figure 8.3b.
- **Convert Binary by removing Probable(1,2):** Figure 8.3c. shows that removing probable index points, 2 levels(1,2) from 4 level output (3,2,1,0) to have 2-level input (3,0).

If we have the confidence in the 4-level index ratings, trying to classify 4-level dataset and converting into binary after the processing may have advantages over other approaches. But we know that probable index points and probable not index points are the options to use when the experts are not sure. So in this case removing probables in the beginning may be a better approach since we are not using uncertain data in our dataset. This may produce a better training set. On the other hand once

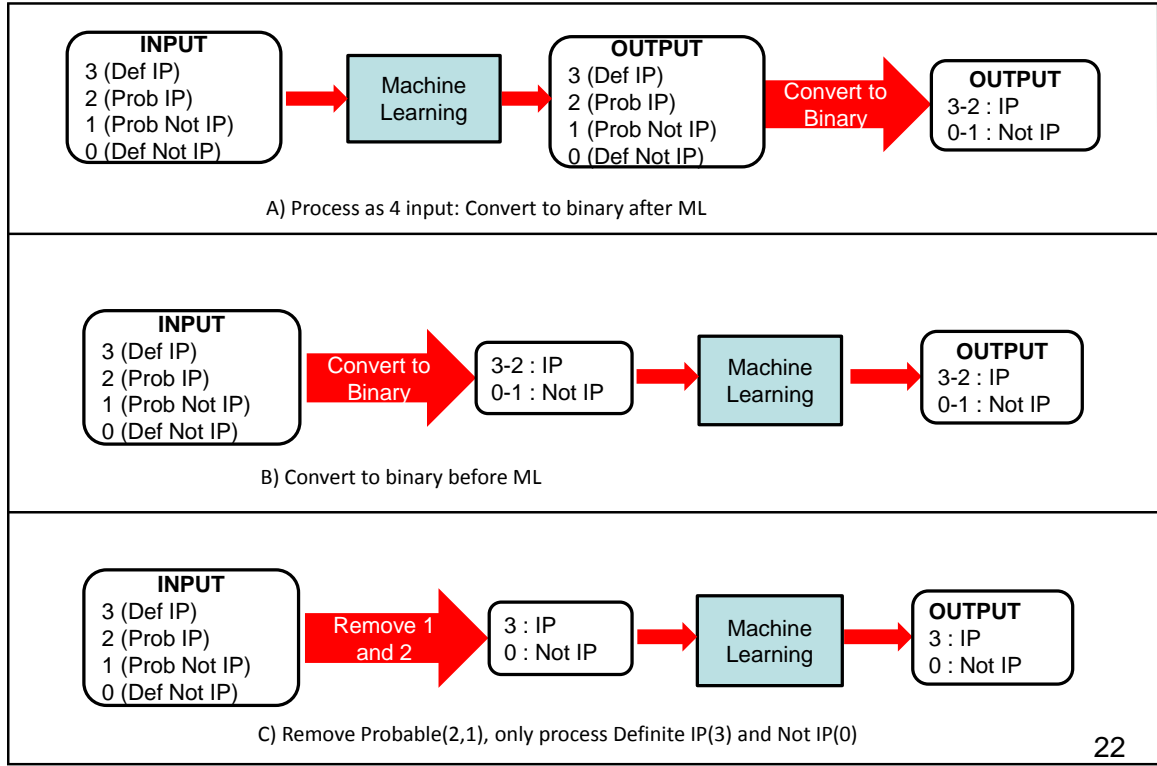


Figure 8.3: Possible strategies to handle 4-level input and 2-level output in machine learning

we remove the 1,2 option from data set we will be losing some training data. To determine which approach is better, we did a small experiment with these three approaches. We divided our dataset into train (60%) and test (40%) set as shown in Figure 8.4. Out of 25 videos only 8 videos had probable index points which is 28% of total dataset. These videos kept in training set and randomly selected rows from other videos were added to training set so that training dataset size become 60%. The rest became our test dataset which has no probable index points and in total it is (40%) of the all dataset.

In the first approach we trained the train dataset without any modifications and

created our model as shown in Figure 8.5. In the second approach we converted the train dataset class values into binary and created the second model. In the third approach, probable index points in the dataset is removed and model is created in converted dataset. All approaches were tested with the same test dataset.

Three different machine learning algorithms used to evaluate these three approaches. Figure 8.6 shows the accuracy of the approaches in selected machine learning algorithms.

Video ID	# of Definitely Not Index Points(0)	# of Probably Not Index Points(1)	# of Probably Index Points(2)	# of Definitely Index Points(3)	Total # of Transition Points
1	53	25	12	9	99
2	43	15	9	3	70
3	36	12	10	5	63
4	2	12	3	2	19
5	24	4	4	7	39
6	23	1	5	2	31
7	22	0	4	1	27
8	99	0	2	6	107
9	22	0	0	8	30
10	78	0	0	4	82
11	41	0	0	4	45
12	46	0	0	5	51
13	72	0	0	8	80
14	99	0	0	5	104
15	49	0	0	5	54
16	78	0	0	6	84
17	110	0	0	3	113
18	112	0	0	5	117
19	59	0	0	6	65
20	71	0	0	7	78
21	55	0	0	6	61
22	79	0	0	8	87
23	55	0	0	9	64
24	27	0	0	4	31
25	24	0	0	3	27
Total	1379	69	49	31	1628

4 level Index Points

28% of Data

60% of Data

R
a
n
d
o
m
i
z
e
d

40% of Data

Figure 8.4: Dividing dataset to train and test to find the best approach for handling 4-level input and 2-level output

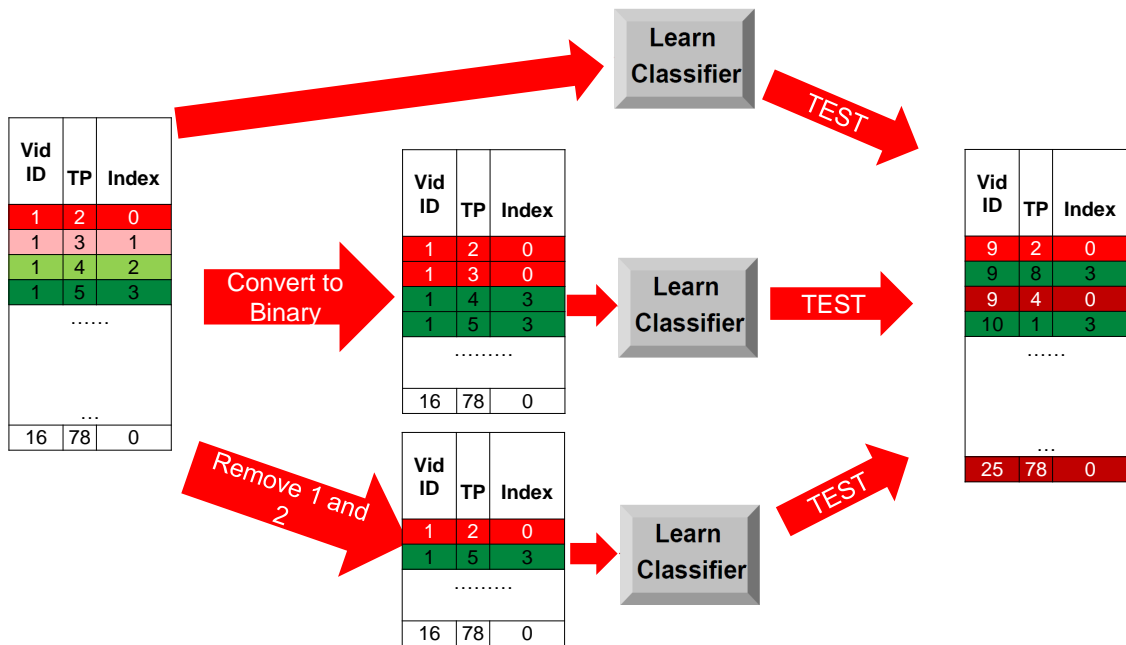


Figure 8.5: Processing traindata and testdata to define best approach to handle 4-level input and 2-level output

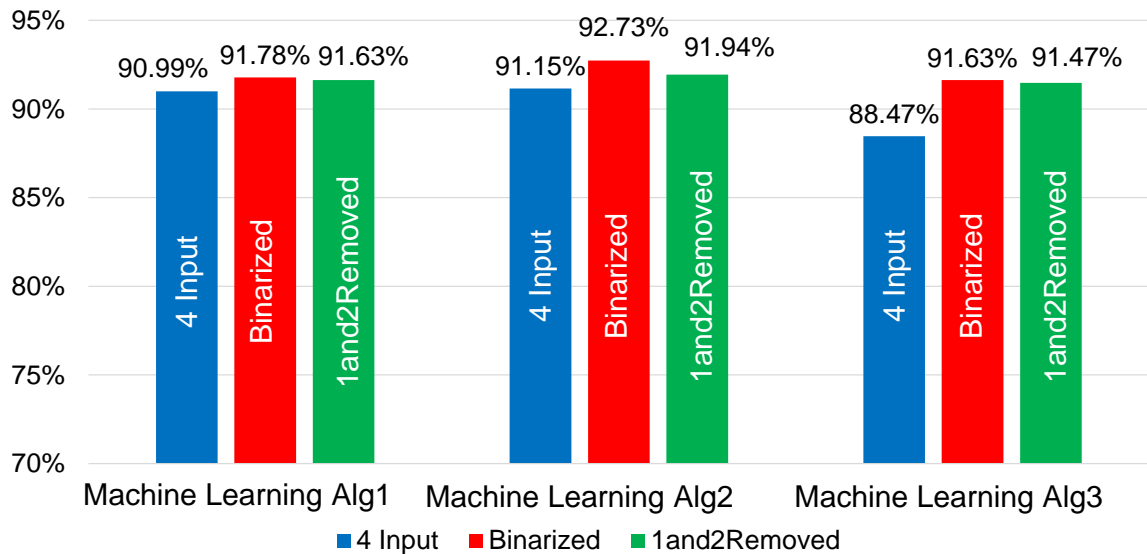


Figure 8.6: Experiment result on train and test dataset for different approaches with various machine learning algorithms: converting to binary performs better in all algorithms

8.4 Need for Number of Index Points Parameter

In the previous section we concluded that the train data should be converted to binary before selecting the classifier and creating the machine learning model. But we also need to make sure that we can use the output of machine learning practically for video indexing. For example a video is expected to have 5-20 index points and those have to be shown to the user in ICS Video player so they can navigate and access the content. If an hour video has 100 transition points and 15 of them are index points, machine learning Indexing should provide number of index points in the range of 5-20. We cannot use directly a 2-3 index points output or 30-50 index points output of machine learning indexing for our system. To figure out this issue we have used the experiment result from the previous section.

The question we are trying to investigate is whether we can use machine learning output as it is shown in Figure 8.7. It turns out we cannot directly use the output machine learning because they give much less number index points than actual number of index points. Figure 8.8 shows the actual number of index points in ground truth and the number of index points provided by machine learning algorithms. It can be seen that in all three different machine algorithms, at least 75 % of index points are not found. This leads us to follow another strategy to apply machine learning for video indexing. As Figure 8.9 shows the number of index points should be provided to machine learning and the output should be filtered accordingly. How can we use machine learning output to produce as many index point as we want. One way is to redesign the current state of a machine learning algorithm in such a way

that it will create that number of index point as an output. But this will require to redesign most of the algorithms and this may cause to loose the benefits which come with those approaches. Another way is to use the probability distribution of classes created by ensemble machine learning algorithms. In other words, if we use ensemble methods we can have the probability distribution for each class index point and not index point to sort the output based on the highest probability value of index point and select the top N rows as index point as shown in Step 7 in Figure 8.10.

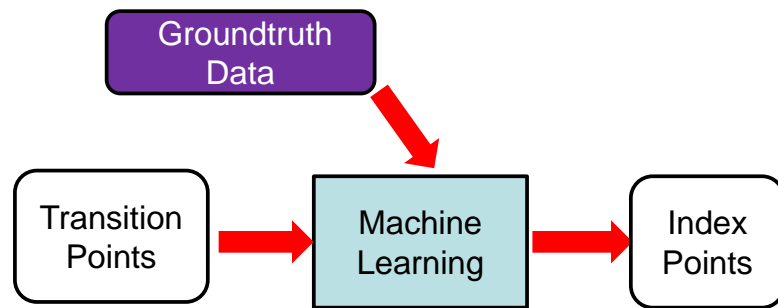


Figure 8.7: Can we do video indexing by using machine learning without providing number of index points parameter

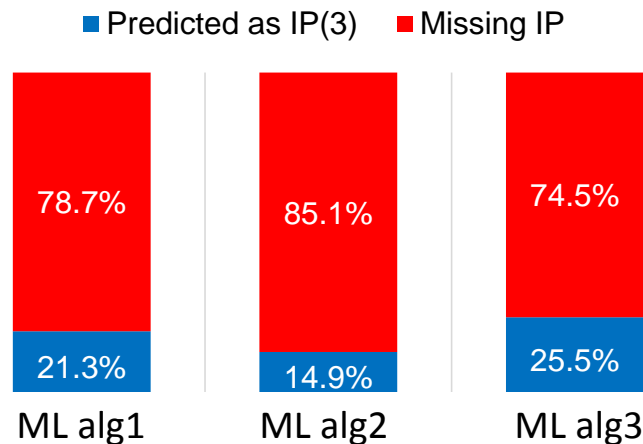


Figure 8.8: Actual number of index points vs findings of machine learning as index points

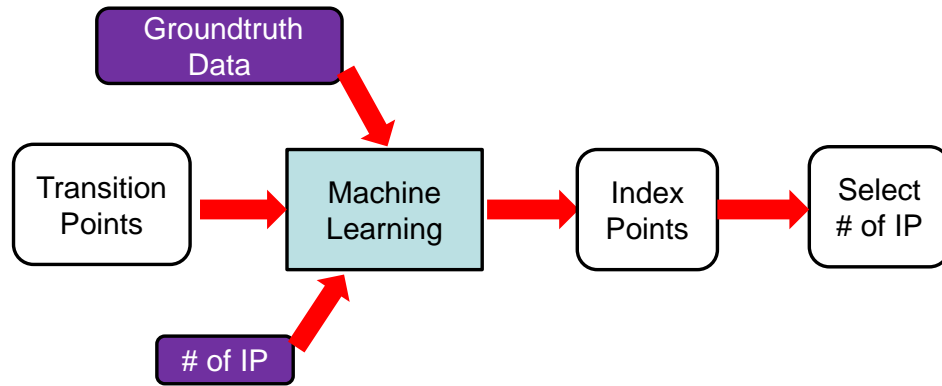


Figure 8.9: Correct strategy to apply machine learning for video indexing: Number of index points should be provided to machine learning and post processing should be done to select desired number of index points.

The procedure for indexing video with a machine learning algorithm is depicted in Figure 8.10. In the first state, ground truth dataset had all transition points. In the second state, first transition points are removed from dataset, because as default they are index points, since beginning of a video is a new topic. They do not have to be trained or predicted. In the third state, index values are converted from 4-level(0,1,2,3) to 2-level(0,1) by changing values of (0,1) to 0 and (2,3) to 1. This converted dataset is the training set for a machine learning so that it can learn and create a model as shown in step 4. After the machine learning model is created, if we want to index a new video, we provide number of index points as a parameter as shown in step 5. The machine learning algorithm will predict which transition points should be defined as index point or not. If we use an ensemble model we will also get probability distribution for each transition point index value. The higher the value of probability distribution for index 1, the more confidence that it is an index value. As shown in step 7, if we need to choose N number of index points, we can choose

the transition points as index points which has the N highest distribution for index value 1. In this example N is defined 3 and the first transition point is index point as default, therefore the other two are chosen based on probability distribution.

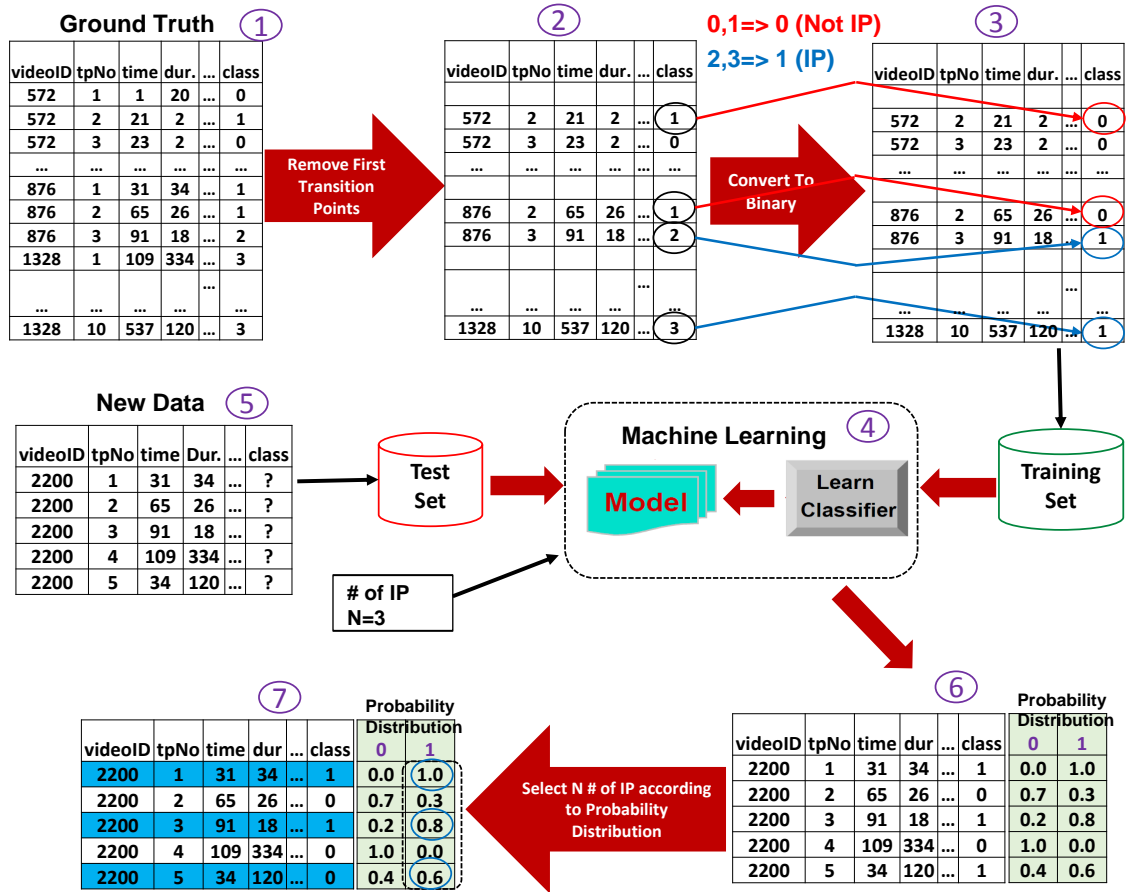


Figure 8.10: Seven steps for indexing video with a machine learning algorithm

8.5 Choosing Machine Learning Algorithms: Ensemble Models

There are many popular machine learning algorithms such as support vector machines, decision trees, bayesnet, among others. These algorithms are available in mostly free and open source libraries with different programming languages: Python, Java, Matlab, C#. Since our whole system was in Java we have used in our system a Java based library, Weka[20], to experiment machine learning approaches. We know that there is no best algorithm that works for all datasets as stated in “No free Lunch Theorem” [51]. We also know that as Turney[50] postulates we choose machine learning algorithms by the following priorities.

- The accuracy, error rate
- Generalization
- Training/Testing time and space complexity
- Interpretability
- Easy programmability

For feasibility purposes in our system, we need to also consider whether the algorithm has probability distribution. which is possible by ensemble models such as Adaboost [17], RandomForest [10] and Bagging [9]. In ensemble models, there are multiple base models, each covers a different part (region) of the input space. Each base model is trained on a slightly different train set. Ensemble model, combines predictions of all

models to produce the output. The goal is to improve the accuracy of the base model. And it is a well known fact that ensemble models generally provides more accuracy than base models. The experiment on our dataset confirms that improvement. Figure 8.11 shows the accuracy rates of some popular machine learning algorithms on full data set with 10-fold cross validation. As it suggests, ensemble models have high accuracy comparing to base models.

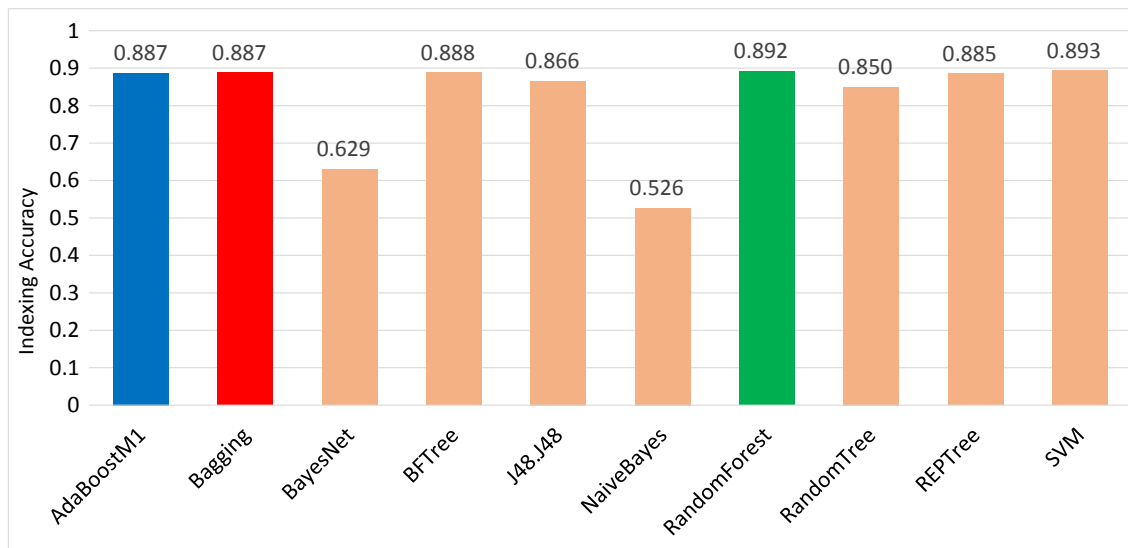


Figure 8.11: 10-Fold Cross Validation results of different machine learning algorithms: high accuracy with Ensemble models AdaboostingM1, Bagging and RandomForest.

Figure 8.12 shows the accuracy rates of ensemble machine learning algorithms on full data set with 10-fold cross validation. As it suggests although the accuracies of these algorithms are very close, AdaboostM1 has very low true positive rate and very low precision comparing to other algorithms. Its high accuracy mainly comes from higher true negative rates and lower false positive rates. This can be observed from the Table 8.1: AdaboostM1 has very low true positives but very high true

negatives. And because the dataset is imbalanced as shown in Figure 8.13, effect of true negatives is higher than effect of true positives. An algorithm which gives all output as not index points can not be used for video indexing but it will still have very high accuracy ($1448/(1448+80)=0,947$). Because of this reason, Bagging and RandomForest is preferable to AdaboostM1 and in the next section only Bagging and RandomForest is being used.

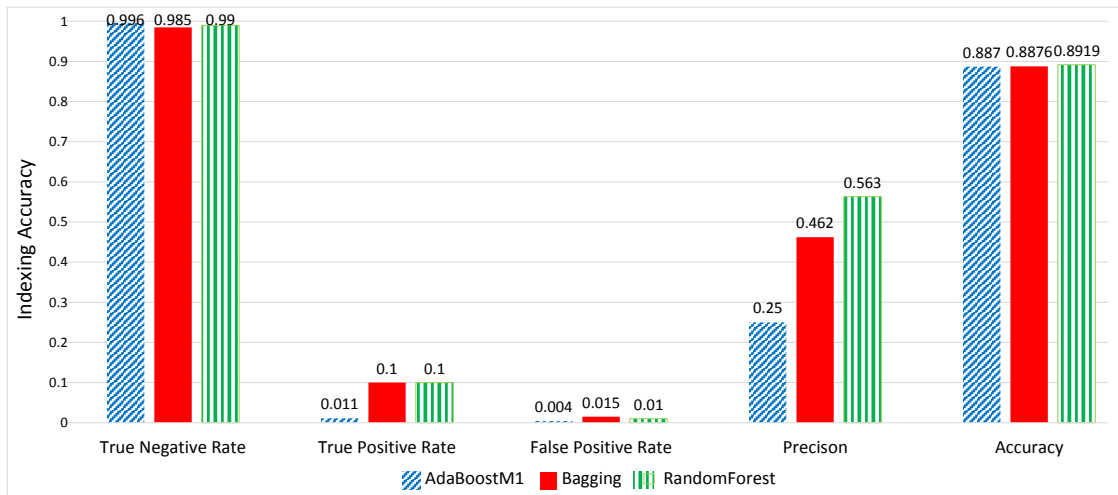


Figure 8.12: 10-Fold Cross Validation results of different metrics for ensemble models: AdaboostingM1 has very low true positive rate and very low precision

Table 8.1: 10-Fold Cross Validation results of different ensemble models: true positive, true negative, false positive, false negative

Algorithm	True Positive	True Negative	False Positive	False Negative
AdaBoostM1	6	1442	178	2
Bagging	21	1427	162	18
RandomForest	14	1434	162	18

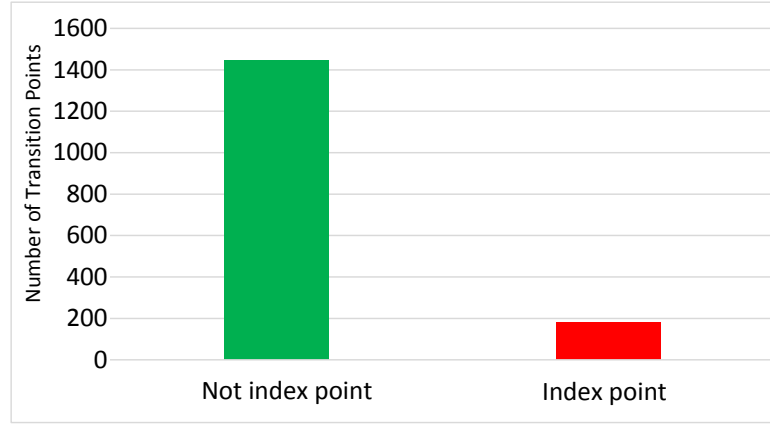


Figure 8.13: Dataset is imbalanced: number of *not index point* is much higher than *index points*

8.6 Attribute Importance by Information Gain

For machine learning experiment, 405 different features are created to use in classification. Attribute importance is calculated by the information gain [52] to see which features are more important or which features provide more information for topic segmentation in videos. The results of information gain calculation is listed as most important 50 features in Table 8.2

According to the ranking of features most important 3 features in order are:

- firstTimeWordsCount: number of words that appear in the video for the first time
- disLeftcos1: cosine similarity of slide with the previous slide
- imgLeftDif: image color difference of slide with the previous slide

It can be observed from Table 8.2 that there are many ngram,title related features

Table 8.2: The most important 50 features based on information gain

Ord	Score	Attribute	Ord	Score	Attribute
1	0.0852	firstTimeWordsCount	26	0.0696	title10left5min
2	0.0845	disLeft.cos1	27	0.0696	title5left5min
3	0.0842	imgLeftDif	28	0.0691	disLeft.cosMin4
4	0.0841	ngrm2Title10left4min	29	0.0691	ngrm2Title5left2min
5	0.0817	ngrm2Title10left3min	30	0.0689	title5left6min
6	0.0804	ngrm2Title10left2min	31	0.0689	title10left6min
7	0.0788	leftCommonWords3Title10	32	0.0683	ngrm2Title10left7min
8	0.0788	leftCommonWords3Title5	33	0.0682	ngrm3left2min
9	0.0776	leftCommonWords2Title10	34	0.0678	ngrm2Title5left5min
10	0.0776	leftCommonWords2Title5	35	0.0662	title5left7min
11	0.0772	title10left3min	36	0.0662	title10left7min
12	0.0772	title5left3min	37	0.0660	ngrm2left3min
13	0.0769	disLeft.cos2	38	0.0657	ngrm2left2min
14	0.0766	title10left4min	39	0.0655	ngrm2Title5left6min
15	0.0766	title5left4min	40	0.0654	ngrm3Title10left4min
16	0.0758	ngrm2Title10left5min	41	0.0653	ngrm3left3min
17	0.0750	disLeft.cosMin3	42	0.0651	ngrm3Title10left3min
18	0.0747	disLeft.cosMin2	43	0.0649	ngrm2left4min
19	0.0743	ngrm2Title5left4min	44	0.0644	ngrm3left6min
20	0.0742	uniqWordsCount	45	0.0644	ngrm2Title10left8min
21	0.0739	allWordsCount	46	0.0640	title10left8min
22	0.0727	ngrm2Title5left3min	47	0.0640	title5left8min
23	0.0721	title5left2min	48	0.0639	ngrm2Title5left7min
24	0.0721	title10left2min	49	0.0634	ngrm2Title10left10min
25	0.0711	ngrm2Title10left6min	50	0.0629	ngrm3left5min

among the most 50 important features which implies that ngrams and large font size are important. It is also clear that most of the important features are related to left, meaning that comparison of a slide with the previous one provides more information than comparing to right. In related to that, it can be seen in Table 8.3 most of the least important features are related to right comparison. This does not mean that right side comparison, comparing a slide with proceedings, is unnecessary but it means that left side comparison provides more information than right side comparison and if the left side comparison is done right side comparison is not

needed since it will not add any more information.

Table 8.3: The least important 50 features based on information gain

Ord	Attribute	Order	Attribute
356	title5right9min	381	ngrm4right4min
357	ngrm3rightAll	382	ngrm4right3min
358	title5right8min	383	ngrm2right10min
359	ngrm2Title5right7min	384	ngrm2right9min
360	ngrm3right7min	385	ngrm2Title5rightAll
361	ngrm3right9min	386	ngrm2right8min
362	ngrm3right6min	387	disRight.cosMin10
363	ngrm3right5min	388	disRight.cosMin9
364	title10right7min	389	disRight.cosMin8
365	title5right7min	390	ngrm3right3min
366	leftDurationAll	391	ngrm2Title5right5min
367	rightDuration1	392	ngrm4right9min
368	ngrm4rightAll	393	ngrm3right10min
369	title10right10min	394	title5right10min
370	title10right9min	395	ngrm3right4min
371	ngrm2right4min	396	disRight.cosAll
372	ngrm2right5min	397	disRight.cosMin1
373	ngrm4right6min	398	disRight.cosMin7
374	ngrm4right7min	399	disRight.cosMin2
375	ngrm4right10min	400	disRight.cosMin6
376	ngrm2right3min	401	disRight.cosMin5
377	ngrm4right8min	402	disRight.cosMin4
378	ngrm4right5min	403	disRight.cosMin3
379	ngrm2right6min	404	ngrm2Title5right6min
380	ngrm2right7min	405	ngrm3right8min

All the features are listed in Table 8.4.

Table 8.4: All features created for machine learning indexing

Attribute Name (Left Group)	Attribute Name (Right Group)	Number of Attributes (Total=405)
leftCommonWords1-3	rightCommonWords1-3	6
leftDuration1-3	rightDuration1-3	6
leftCommonWordsAll	rightCommonWordsAll	2
leftDurationAll	rightDurationAll	2
leftMin1-Min10	rightMin1-Min10	20
disLeft.cos1-3	disRight.cos1-3	6
disLeft.ecl1-3	disRight.ecl1-3	6
disLeft.jac1-3	disRight.jac1-3	6
disLeft.dic1-3	disRight.dic1-3	6
disLeft.cosAll	disRight.cosAll	2
disLeft.eclAll	disRight.eclAll	2
disLeft.jacAll	disRight.jacAll	2
disLeft.dicAll	disRight.dicAll	2
disLeft.cosMin1-Min10	disRight.cosMin1-Min10	20
disLeft.eclMin1-Min11	disRight.eclMin1-Min11	20
disLeft.jacMin1-Min12	disRight.jacMin1-Min12	20
disLeft.dicMin1-Min13	disRight.dicMin1-Min13	20
ngram2LeftMin1-Min10	ngram2Right1min-10min	20
ngram2LeftAll	ngram2RightAll	2
ngram3LeftMin1-Min10	ngram3Right1min-10min	20
ngram3LeftAll	ngram3RightAll	2
ngram4LeftMin1-Min10	ngram4Right1min-10min	20
ngram4LeftAll	ngram4RightAll	2
firstTimeWordsTitle5Count	firstTimeWordsTitle10Count	2
leftCommonWords1-3Title5	rightCommonWords1-3Title5	6
leftCommonWords1-3Title10	rightCommonWords1-3Title10	6
title5Left1Min-10Min	title5Right1Min-10Min	20
title10Left1Min-10Min	title10Right1Min-10Min	20
ngram2Title5Left1Min-10Min	ngram2Title5Right1Min-10Min	20
ngram2Title5leftAll	ngram2Title5RightAll	2
ngram2Title10Left1Min-10Min	ngram2Title10Right1Min-10Min	20
ngram2Title10leftAll	ngram2Title10RightAll	2
ngram3Title5Left1Min-10Min	ngram3Title5Right1Min-10Min	20
ngram3Title5leftAll	ngram3Title5RightAll	2
ngram3Title10Left1Min-10Min	ngram3Title10Right1Min-10Min	20
ngram3Title10leftAll	ngram3Title10RightAll	2
ngram4Title5Left1Min-10Min	ngram4Title5Right1Min-10Min	20
ngram4Title5leftAll	ngram4Title5RightAll	2
ngram4Title10Left1Min-10Min	ngram4Title10Right1Min-10Min	20
ngram4Title10leftAll	ngram4Title10RightAll	2
imgLeftDif	imgRightDif	2
allWordsCount, uniqWordsCount, firstTimeWordsCount		3

Chapter 9

Evaluation and Experimental Results

9.1 Indexing Accuracy

To determine the improvement achieved by text-based indexing methods and machine learning indexing over non-text-based method, the set of videos in ground truth are processed by each approach separately. The types of text-based indexing algorithms evaluated are Fixed Grouping, Linear Weighted, Non-linear Weighted, and Boundary-based. RandomForest and Bagging are evaluated as machine learning indexing algorithms. The ideal case is the theoretical output where the index points selection is from the ground truth provided by the instructor. Ideal output achieves the best possible accuracy by manually picking the required number of index points out of the transition points marked as index points in the ground truth data. The

required number of index points may not necessarily match the total number of index points in the ground truth. Therefore, the ideal case may not achieve 100% accuracy. The required number of index points for the algorithms was calculated from different duration of index point values for 6, 8, 10, and 12 minutes. The performance comparison is based on the average accuracy score for the given set of videos at various number of index points. The following Figure 9.1 to 9.4 provide the average accuracy achieved for each indexing algorithm at different number of index points per different time intervals. These charts are also combined in Figure 9.5 to show the relations of different number of index points accuracies in one graph.

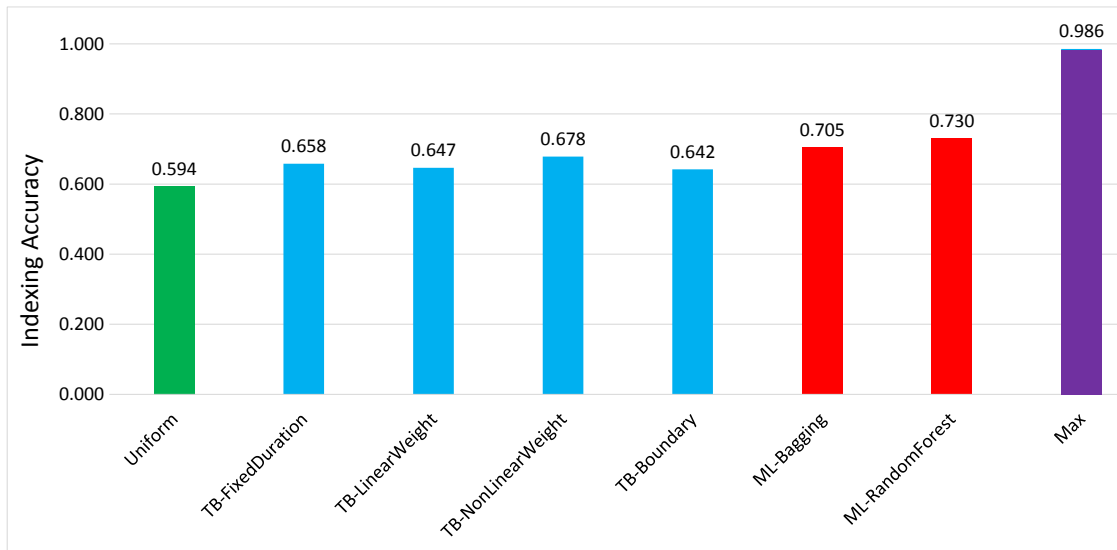


Figure 9.1: Indexing (4-point metric) average accuracy for 25 videos; index point per 6 minutes

It is evident from the average accuracy charts that the text-based indexing methods performed better than non-text-based algorithm and machine learning indexing

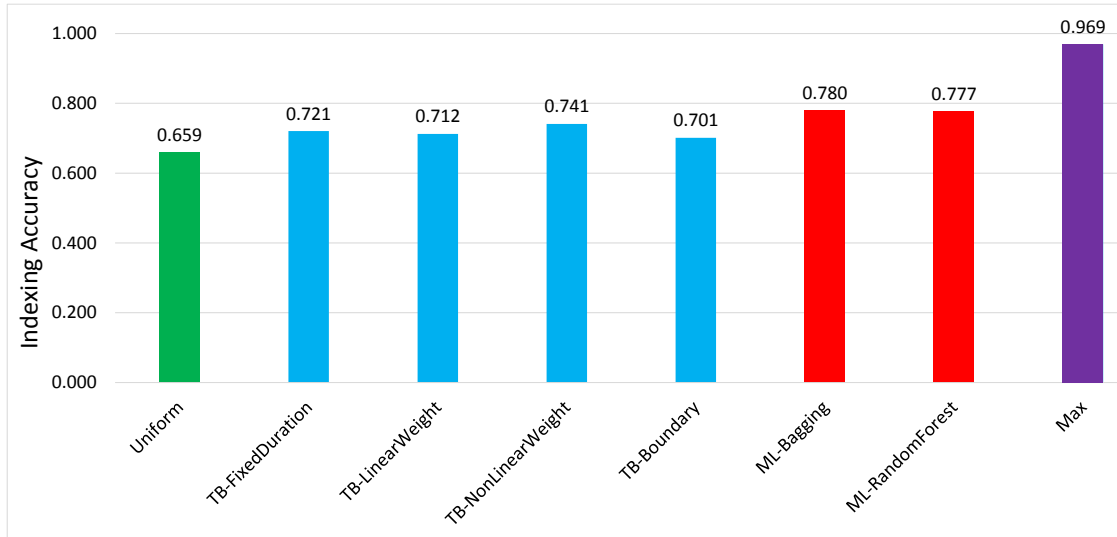


Figure 9.2: Indexing (4-point metric) average accuracy for 25 videos; index point per 8 minutes

algorithms performed better than both. Fixed Duration algorithm performed better than the Uniform algorithm, whereas the Linear Weighted algorithm performed similar to Fixed Duration algorithm. The Non-linear Weighted algorithm performed better than other text-based algorithms. The Boundary-based algorithm fared worse than other text-based indexing algorithms.

The number of required index points is inversely proportional to the value of minutes per index point. For an hour long video, algorithms will find 10 index points if the input selected as index points per 6 minutes, whereas they will find 5 index points if the input is selected as index point per 12 minutes. Thus, when going from index points per 6 minutes to 12 minutes, required number of index points will be reduced. The possibility for algorithms to make mistakes will be reduced as well. This can be observed in Figure 9.5. If the required number of index point per

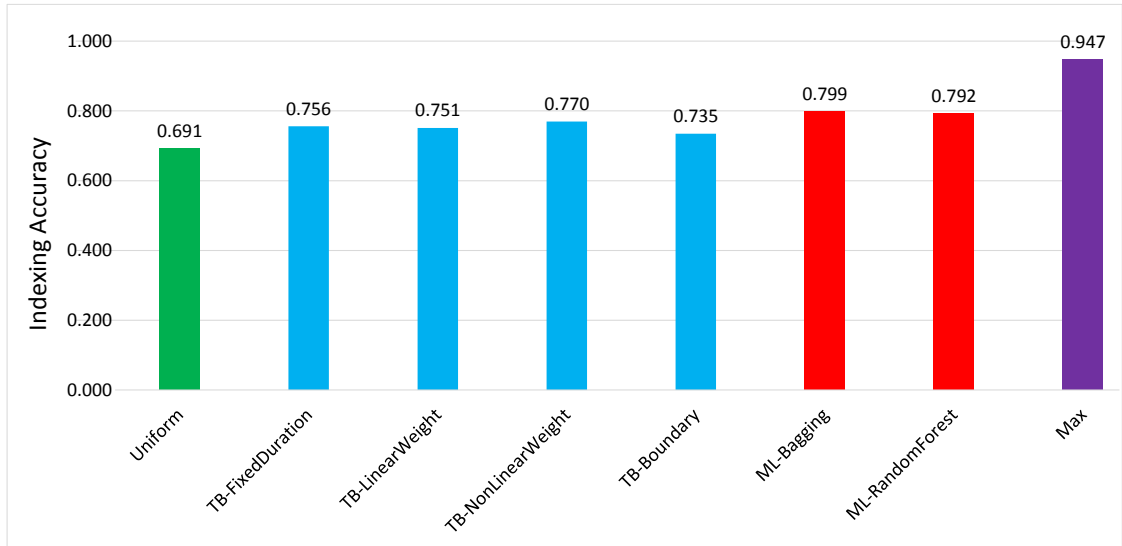


Figure 9.3: Indexing (4-point metric) average accuracy for 25 videos; index point per 10 minutes

minute increased, all the algorithms' accuracies gradually increase and the differences between the algorithm decreases. This is an expected change as the required number of index points set to zero, accuracy of the all algorithms will be same. In contrast, the maximum accuracy that any algorithm can get is decreased as we move from per 6 to 12 minutes, which means that correct average number of index points is around an index point per 6 minutes.

The machine learning algorithms, RandomForest and Bagging accuracies are very close. For index point per 6 minutes, RandomForest is slightly better than Bagging. In contrast, for index point per 8,10 and 12 minutes, Bagging performed better. Using a parameter index point per 6 minutes is more realistic than other intervals since the highest ideal maximum accuracy can be achieved by 6 minutes interval(0.986 vs 0.969, 0.947, and 0.924). As a result of this fact and the RandomForest performed

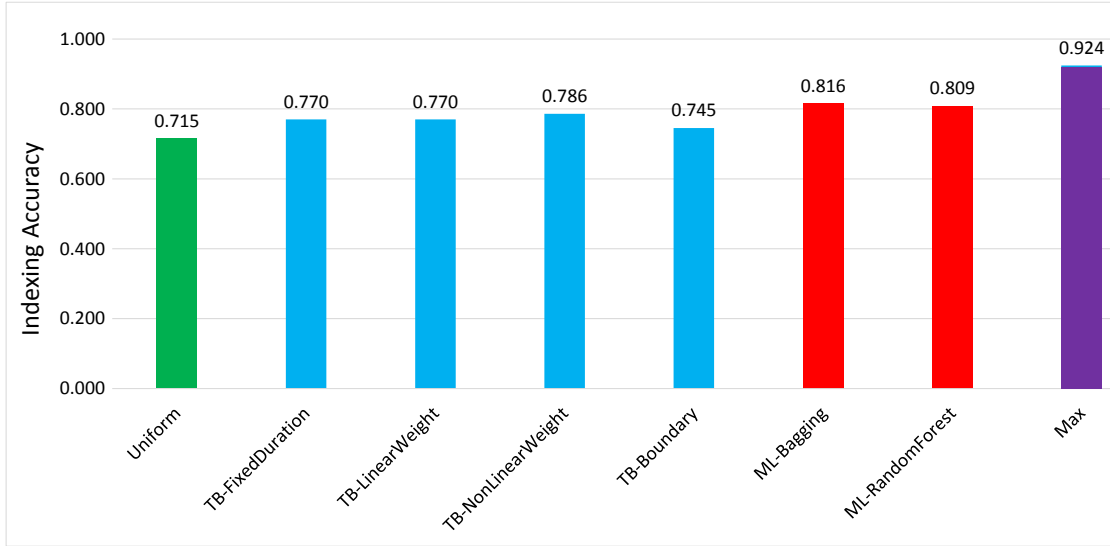


Figure 9.4: Indexing (4-point metric) average accuracy for 25 videos; index point per 12 minutes

better than Bagging in 6 minutes interval, RandomForest is preferable to Bagging for practical purposes. Nonetheless, to clarify which algorithm performs better, another evaluation is done. Instead of fixing number of index point per duration, desired number of index points are given equal to the number of index point in ground truth. Average accuracy results for all videos in this evaluation are displayed in Figure 9.6 with 4-point metric and in Figure 9.7 with 2-point metric. Both results show that RandomForest has slightly higher accuracy than Bagging. It can also be observed that, the accuracy values in 4-point metric (between 0.667-802) are less than 2-point metric (between 0.824-884). And the differences between the accuracies of algorithms are reduced in 2-point metric. This is because in 4-point metric scoring, the accuracy score drops in a nonlinear fashion with errors. In 2-point metric scoring, precision and recall values are also valuable indicators of accuracy and in this evaluation since we provide the exact number index points to the algorithms

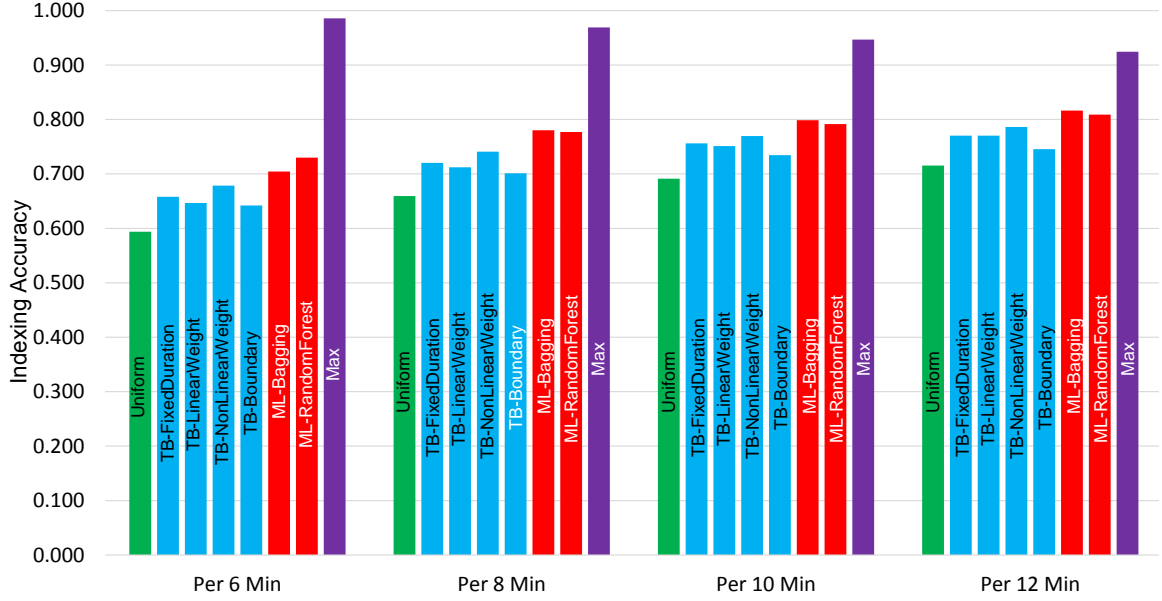


Figure 9.5: Indexing (4-point metric) accuracy for index point per different time interval for all videos

equal to ground truth, precision and recall values are equal (number of false positives are equal to number of false negatives). The accuracy of uniform indexing in 2-point metric scoring is high 0.824. But we should remember from Chapter 8.5 that the data is imbalanced; there are 180 index points and 1448 not index points. Even if an algorithm could not detect any of the index points correctly, and if we provide correct number of index points, it will result in $(1448-180)/16268=0.788$ accuracy score. But it will have 0 precision and recall value. As can be observed in Figure 9.7, uniform indexing has very low precision and recall value, 0.21.

The results from the experiments with the number of index points per different time intervals and with the number of index points equal to number of index point in ground truth indicates that machine learning and text-based indexing algorithms can produce better results than non-text-based methods in achieving topic-based

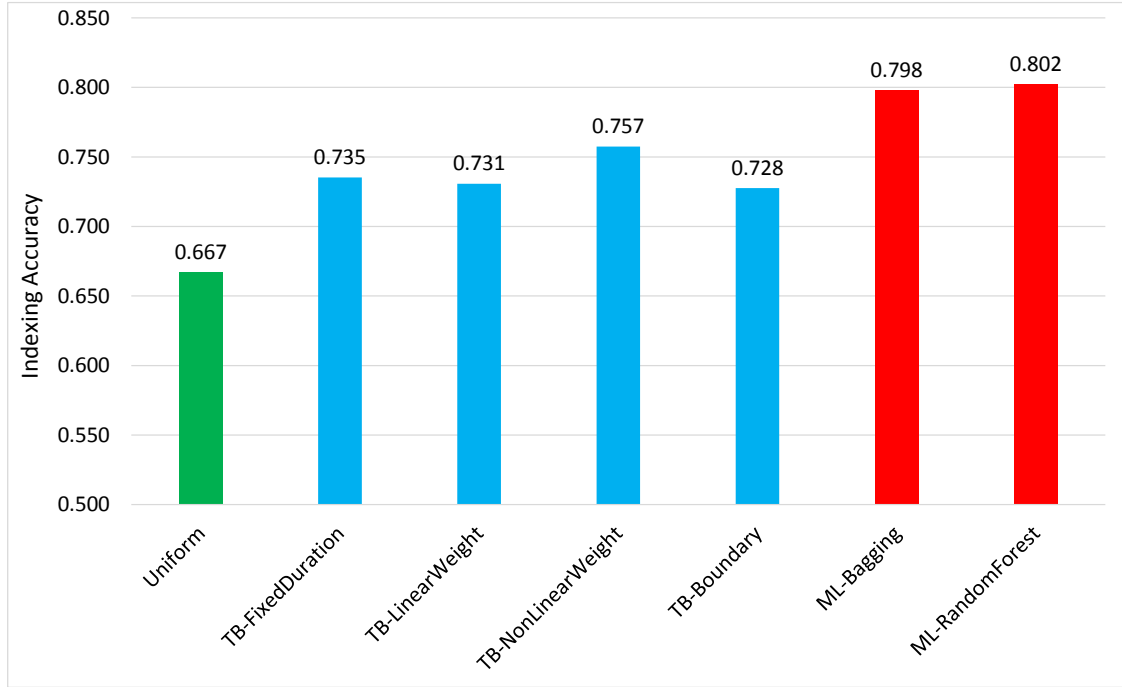


Figure 9.6: Indexing (4-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in ground truth

segmentation. Nevertheless, both results depend on number of index points. It is a fact that the actual number of index points are unknown and a fixed number of index point will not work for all videos. In other words, each video will have different number of index points per hour. 25 videos in the dataset is sorted based on video duration and the number of definitely and probably index points and the total number of transition points is plotted in Figure 9.8. As can be observed that the number of index points and the total number of transition points is not correlated to video duration. In other words, number of index points cannot be predicted by video duration. In this regard, an evaluation which is free from desired number of index points will be a better accuracy metric for the current video indexing algorithms. The

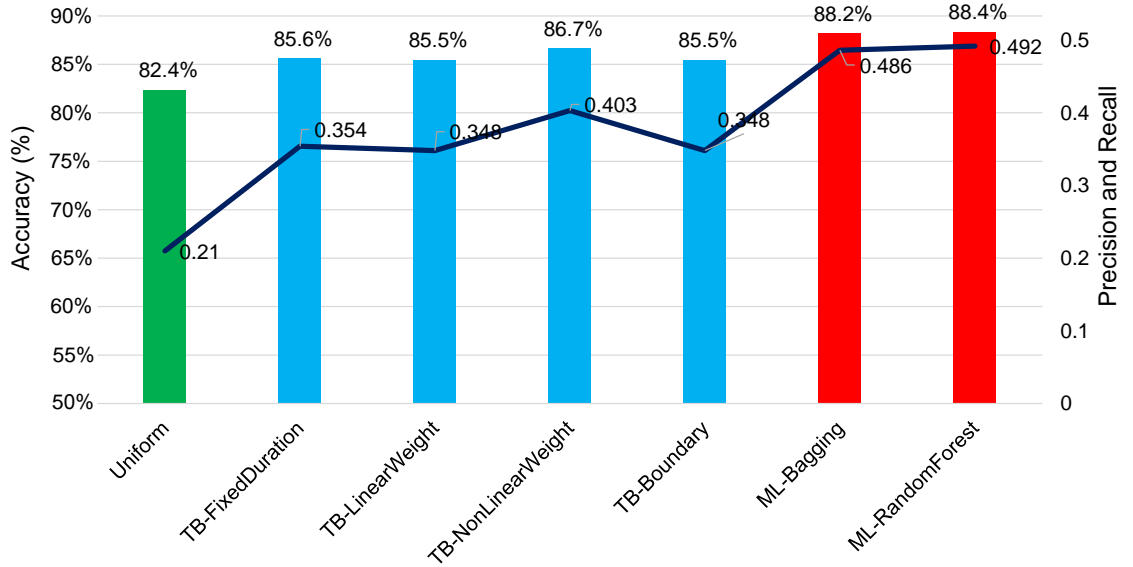


Figure 9.7: Indexing (2-point metric) average accuracy for 25 videos; the number of index points provided to algorithms as in ground truth

Sorting metric as explained in Chapter 6.2.3 is created for this purpose. Ground truth is sorted based on the index value, machine learning algorithms RandomForest and Bagging is sorted by the probably distribution values of index point as explained in Chapter 8. Unfortunately, among the text-based indexing algorithm only Boundary-based algorithm can be sorted without using any index point parameter. Therefore, only Boundary-based indexing algorithm could be compared to machine learning indexing algorithms. The results from this comparison are shown in Figure 9.9. It is clear that the accuracy order is the same with the previous comparisons, from best to worst; RandomForest, Bagging and Boundary. Another observation is that the difference between Boundary text-based indexing and machine learning algorithms are higher in sorting metric than the difference in previous comparisons.

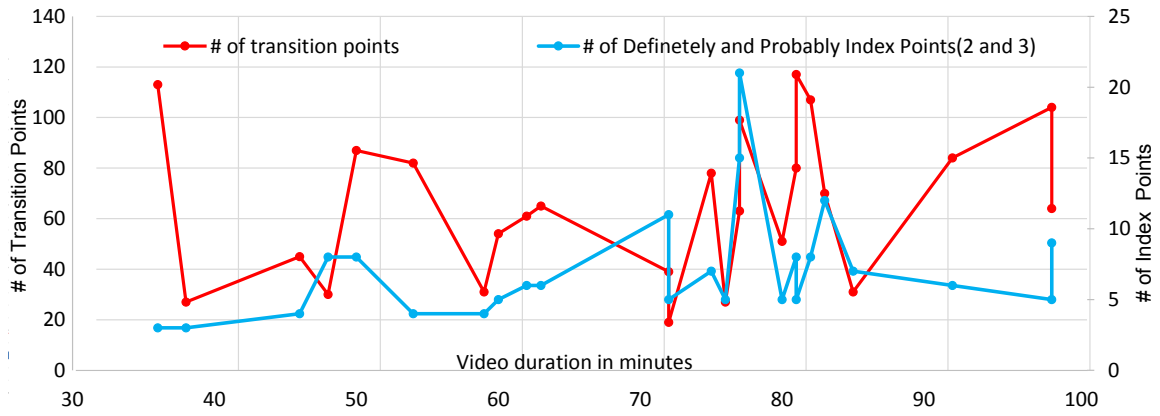


Figure 9.8: Number of transition points and number of index points based on video duration: not linearly correlated

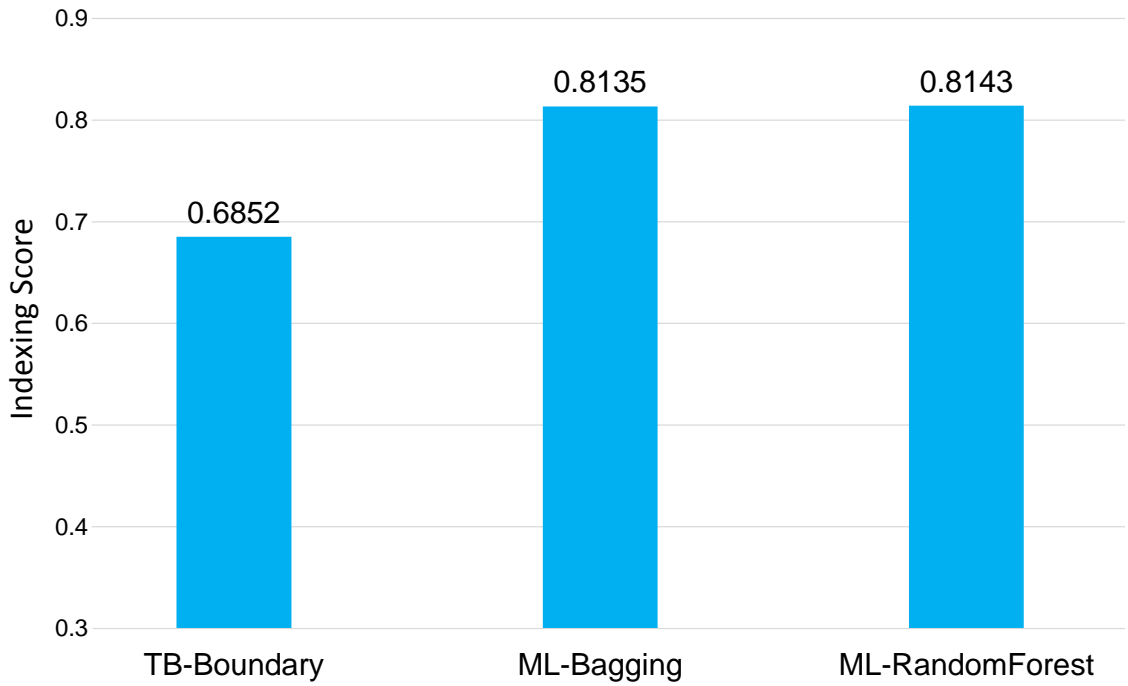


Figure 9.9: Sorting metric indexing score of TB-Boundary, ML-RandomForest, and ML-Bagging

9.2 Survey Results

Indexed Captioned Searchable (ICS) Video usage is assessed to develop an understanding of the overall perceived value of the video lectures and the value of video indexing and keyword search. Surveys were administered over 5 years in more than 10 semesters [6]. Figures 9.10 and 9.11 show the response of approximately 120 students from Spring 2013 and Fall 2013 semester to a required-answer question about the usefulness and value of the indexing. Figure 9.10 shows that 96% of respondents agreed the video indexing was helpful, 96% agreed that the placement of index points in the video timeline was appropriate for the lectures, 95% believe that the layouts of the index images made the index feature easy to use, and 93% agreed that the index points separated a lecture into logical segments. In this figure “Disagree strongly”, “Disagree” and “Disagree slightly” is merged to “Disagree***” due to the low number of responses

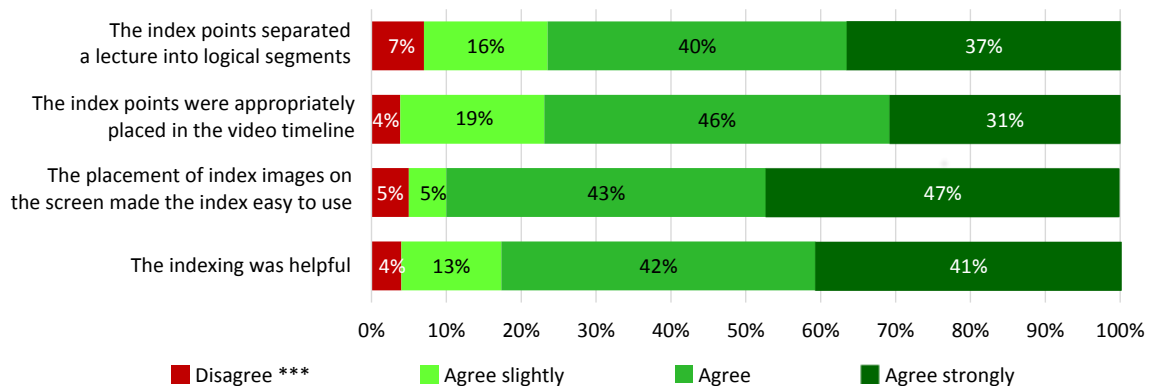


Figure 9.10: Value of video indexing with first set of questions

Responses to additional questions on the value of indexing are presented in Figure 9.11. Students are strongly supportive of the statements that the index feature

functioned well, that the index points provided enough information to identify video segments of interest, and that the index made it easy to navigate the video. The statement that index points represented the start of a new subtopic had somewhat weaker support than the other assertions (10% of students said “hardly ever”, 21% said “sometimes”, 41% said “most of the time” and 28% said “always”). It is important to note that even imperfect indexing is perceived as very valuable by the students.

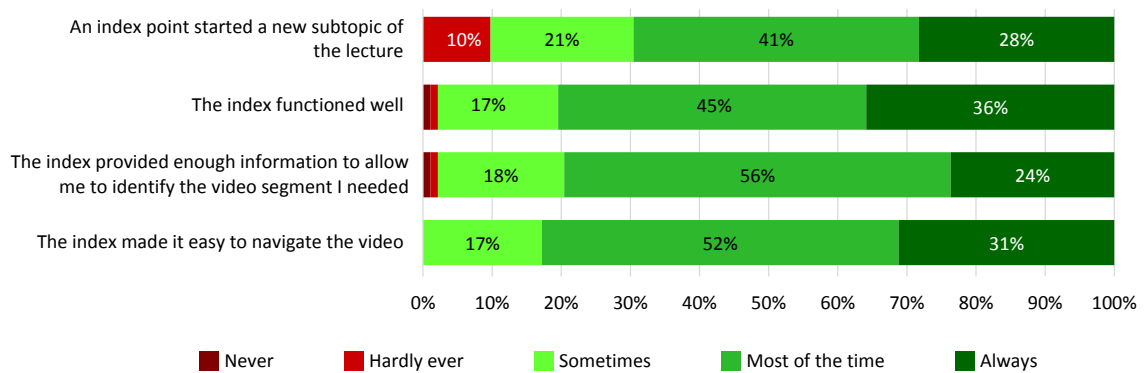


Figure 9.11: Value of video indexing with second set of questions

Figures 9.12 and 9.13 show the responses to the questions on the keyword search. The response rate was low for this set of questions; only 39 students responded. We believe there are several reasons for this. Many students may not see a need for using the keyword search feature as indexing allows navigation of topics inside a video. Index points are clearly visible when the ICS Video player is active and navigation only requires clicking on the index snapshots. In order to utilize the search feature, the user needs to identify the search box and identify and type the search keywords. Also, the exact functionality of the search box may not be obvious to some students and earlier versions of the player had the search box located in a

corner that was not conspicuous.

Nonetheless, of the 39 students who used it, 94 percent of respondents reported that the search feature was easy to use, 81 percent thought the results were appear to be true. While 70 percent felt that feature helped them find the part of the video they intended to find most of the time, only 75 percent reported that they usually knew which words to enter into the search box to find the segment of video they wanted. We speculate that if instructors can increase their students familiarity with the proper contentand thus vocabularythe percentage of successful use to find the intended clip should increase.

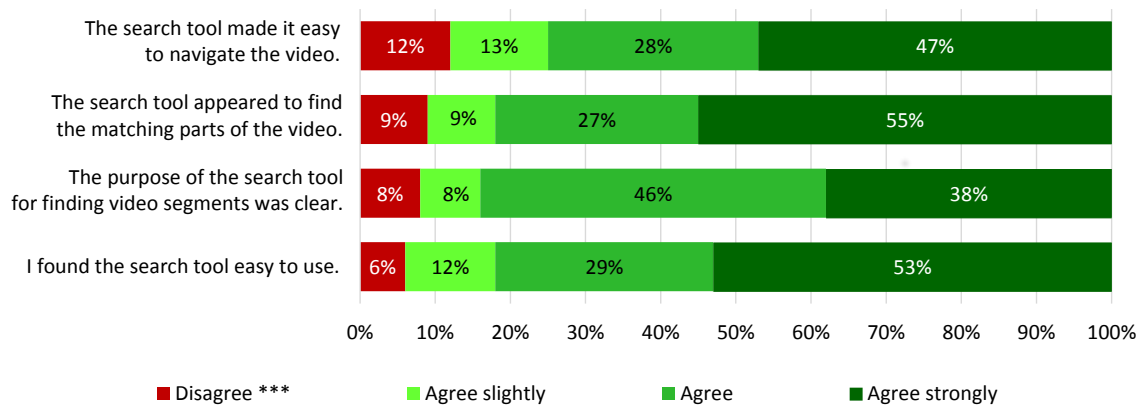


Figure 9.12: Value of search with first set of questions

Additional results in Figure 9.13 show that, at least some of the times, 88 percentage of students (or more, depending on the item) found the search tool helpful, found that the results of the search feature were relevant to what they were looking, knew which words to enter in the search box to find sections, and thought the search tool helped them to find the part of the video they were looking for.

In summary, the results show that keyword search was found to be very valuable

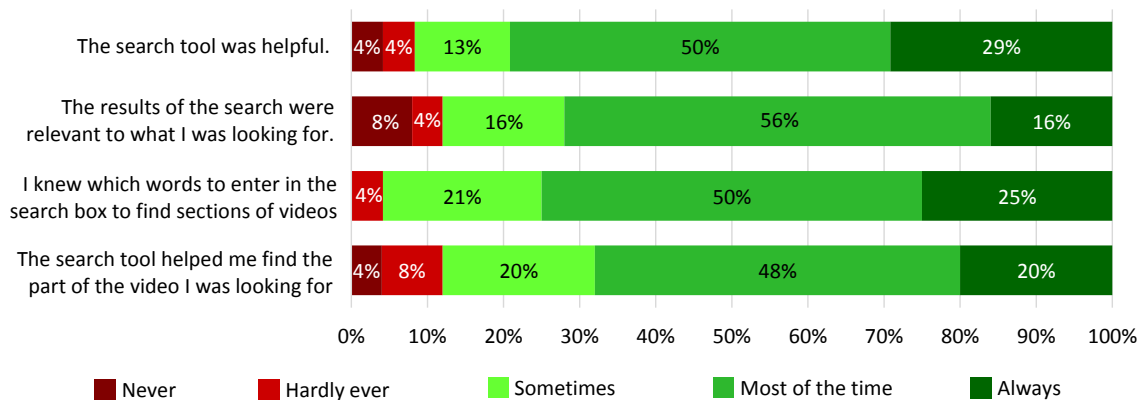


Figure 9.13: Value of search with second set of questions

by the students who used it. However, underutilization remains a problem because the feature may not be needed by all students. Additional training and familiarity with the search feature is needed for wider adoption.

In open-ended comments, students reported several benefits from using the index including (a) saving time, for example one student wrote, “I did not have to wade through the rest of the lecture just to answer one question”; (b) skipping through material the student was familiar with to get to the challenging sections; and (c) returning to a section of the lecture if an interruption occurred. For example, one student wrote, “Sometimes I would have to pause the lecture to take care of other responsibilities that I had to attend to, and when I was ready to come back to the lecture I’d pick up exactly where I was at. It was great!”. Another student said, “The indexing feature, in my opinion, is one of the best parts regarding this video player. It separated the lecture into reasonably sized sections and made it easy to know where to pick a lecture back up if I had to stop watching for a while.”

The data collection instruments for these survey questions are listed in Table 9.1.

Table 9.1: Survey Map: video usage, problems, indexing and search value

Survey Item	Response Categories
Did you have problems viewing one or more videos? Please indicate how many times this semester that you had technical problems in downloading, viewing, or hearing the video?	None 1, 2-3, 4-5, 6-7, 8 or more
Technical problems: Please help us to fix technical problems by describing them to the best of your ability.	Open-ended
The video player includes index points. An example is shown in the image to your right. Clicking on any of the frames in below the video allowed you to go directly to different segments of video. Did you use the index points to access different video segments?	No, yes, don't remember
Please express your agreement or disagreement: The index was helpful. I immediately understood what to do with the index. The placement of index images on the screen made the index easy to use. The index points separated a lecture into logical segments.	Disagree/agree (6-point Likert with don't know as 7th choice)
Please indicate how often you experienced the following: The index provided enough information to allow me to identify the video segment I needed. The index made it easy to navigate the video. The index functioned well. An index point started a new subtopic of the lecture.	Never, hardly ever/seldom, sometimes, most of the time, always
How can we improve the index feature?	Open-ended
The video player includes a search tool so you can search for individual words shown in the video (e.g., words on a slide). You type a word in the search box and if the word is found, one or more index points shows up under the video and you can click on them. Did you use the search tool to search for keywords in any of the lecture videos you viewed?	No, yes, don't remember
Please express your agreement or disagreement: I found the search tool easy to use. The search tool appeared to find the matchings part of the video. The purpose of the search tool for finding video segments was clear. The search tool made it easy to navigate the video.	Disagree/agree (6-point Likert with don't know as 7th choice)
Please indicate how often you experienced the following: The search tool helped me find the part of the video I was looking for. I knew which words to enter in the search box to find sections of videos. The results of the search were relevant to what I was looking for. The search tool was helpful.	Never, hardly ever/seldom, sometimes, most of the time, always
How can we improve the search feature?	Open-ended

Chapter 10

Limitations and Discussions

In the previous chapter, it was concluded that the text-based indexing algorithms performed better than non-text-based algorithm and machine learning indexing algorithms performed better than text-based indexing algorithms by having up to 80% average accuracy. This results show that improvement to the current algorithms can be done, theoretically speaking, there is a 20% difference between these algorithms and a perfect indexing algorithm. Nonetheless, achieving this 20% improvement is not practical because of the diversity of videos and the ambiguity in the ground truth.

Two people who were familiar with the course contents created two different ground truths for 10 videos. The 4-point metric indexing accuracy was calculated from these ground truths with the instructor ground truth. As can be observed from the results in Figure 10.1, in average, accuracies are very close, 0.750 vs 0.762 but each person creates ground truth differently in individual videos. The results shows

that although further enhancements could improve the performance of video indexing algorithms, the performance gains are not expected to reach the ideal output because of the uncertain nature of the ground truth. Noting this limitation, error analysis is done in the following sections for possible further enhancements.

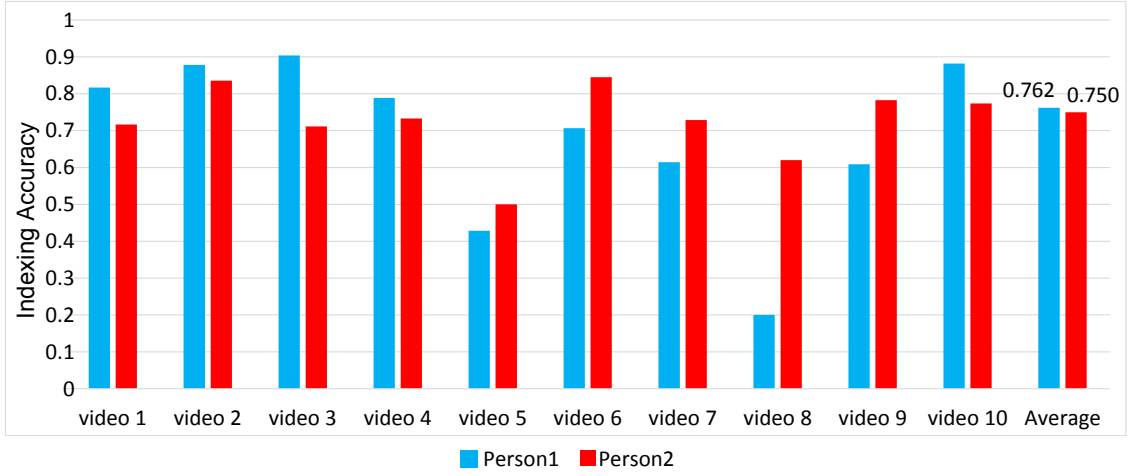


Figure 10.1: Different ground truths and indexing (4-point metric) average accuracy of 10 videos: each person marks differently

The analysis of the outputs of text-based algorithms and machine learning algorithms are done to determine the probable causes for the errors. Undetected index points, false negatives, in the ground truth are examined and listed in the Table 10.1. For this analysis, the required number of index points from the algorithm for each video was set equal to the number of index points in the ground truth. 180 out of 1628 total transition points were marked in the ground truth as probably or definitely index points. As listed in Table 10.1, 20 of the 180 index points were not detected by any of the text-based (TB) or any machine learning (ML) algorithm. As

Table 10.1: Distribution of undetected index points: false negatives

None of TB and ML	None of TB	None of ML	None of TB but All ML	None of ML but All TB
20	60	51	33	9

the table shows, the machine learning approach missed less index points than text-based indexing algorithms, 51 vs 60. ML could detect 33 out of 60 index points that were not detected by any TB algorithm. Though there are 9 index points that were not detected by any ML but all the TB algorithms did.

10.1 Index Points Which are not Detected by Any Algorithm

Each of the index points that were not detected by any algorithms is examined and cause of errors is identified. Figure 10.2 shows the distribution of the causes. 72% of errors occurred because of the slides that loaded by animation or the slides that did not have enough text. The rest of the errors were because of the outline slides and similar slides in topic. Each of these errors are examined in the following.

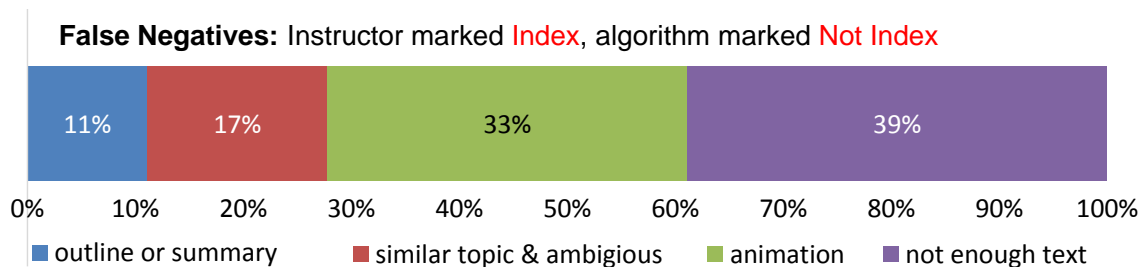


Figure 10.2: Distribution of cause of errors for index points that are not detected by any text-based and machine learning algorithm

10.1.1 Outline or Summary Slide in a Hierarchical Lecture Organization

On rare occasions, the presence of an outline slide that outlines the lecture organization can cause false negatives or wrong index point selection as depicted in Figure 10.3. This is because the outline slide gives an overview or bullet points of the sub-topics in the lecture. The instructor may consider each sub-topic as a separate index point. However, because of the presence of some amount of text in the outline slide, the actual topic segment merges to the outline slide because of relatively high similarity to the outline slide, thus causing false negatives.

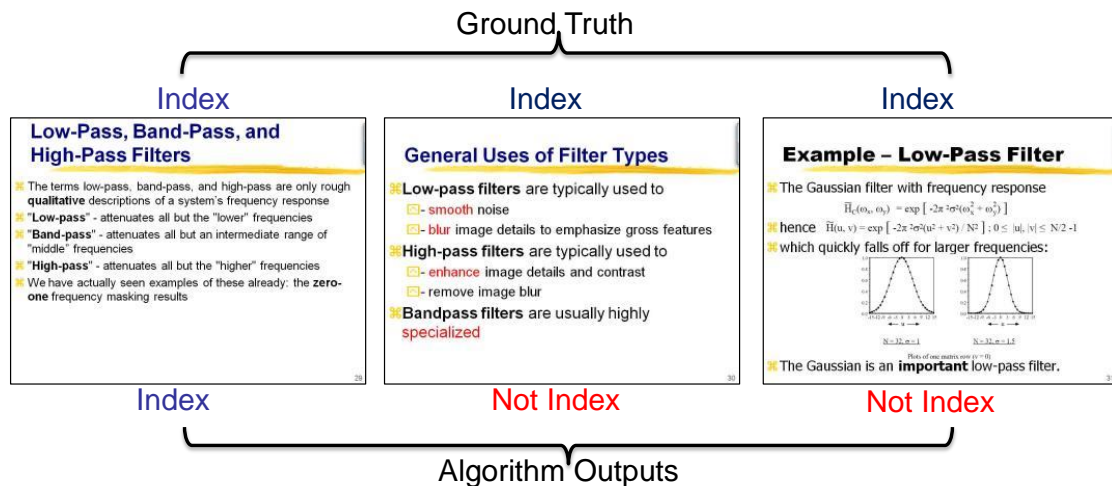


Figure 10.3: Outline slide in a hierarchical lecture organization

10.1.2 Similar to Previous Topic: Ambiguous

When the new topic is very similar to the previous topic, chances are high for such segments to merge with the previous topic segments, thus causing false negatives.

Those index points are also identified by instructors as probably index point because of their ambiguity. In Figure 10.4, the second and third slide is marked as probably index point by instructor. Nevertheless, due to the high text similarity they both merged to first slide and only the first one selected as index point

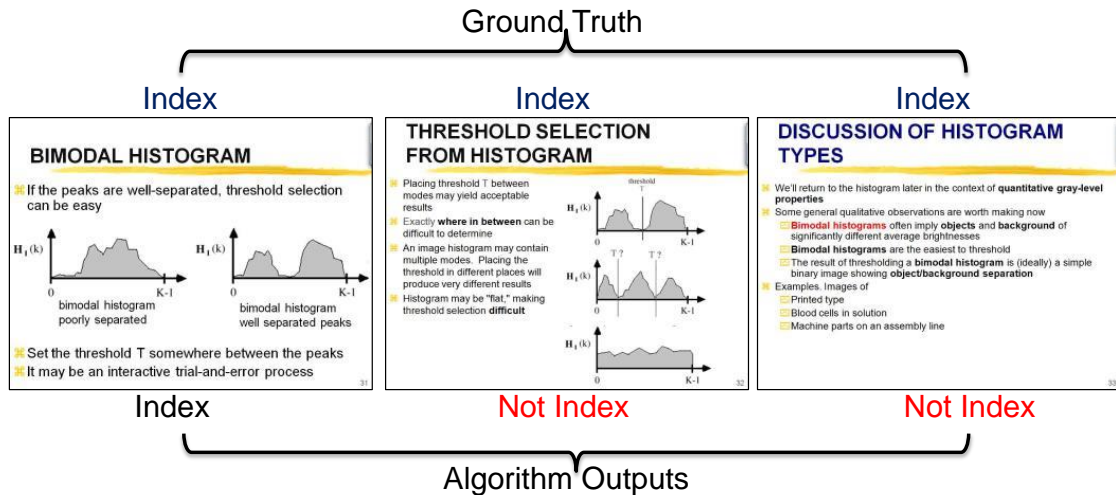


Figure 10.4: Consecutive slides have very similar topics

10.1.3 Animations and Slide Transition Effects

The lowest indexing accuracy comes from the videos having animations, slide transition effects. One of the common animation used is continuing the previous slide text by adding new lines, as shown in Figure 10.5. Although the second and third image is the same slide, they are marked as different transition points due to the high image color difference. The new topic starts with the second transition point but because the third one has smaller duration and merges to fourth slide, algorithm finds the third one as index point instead of second one. If the further merging

happened, third and fourth slide could be merged to second slide but this does not happen because the desired number of index points is already reached.

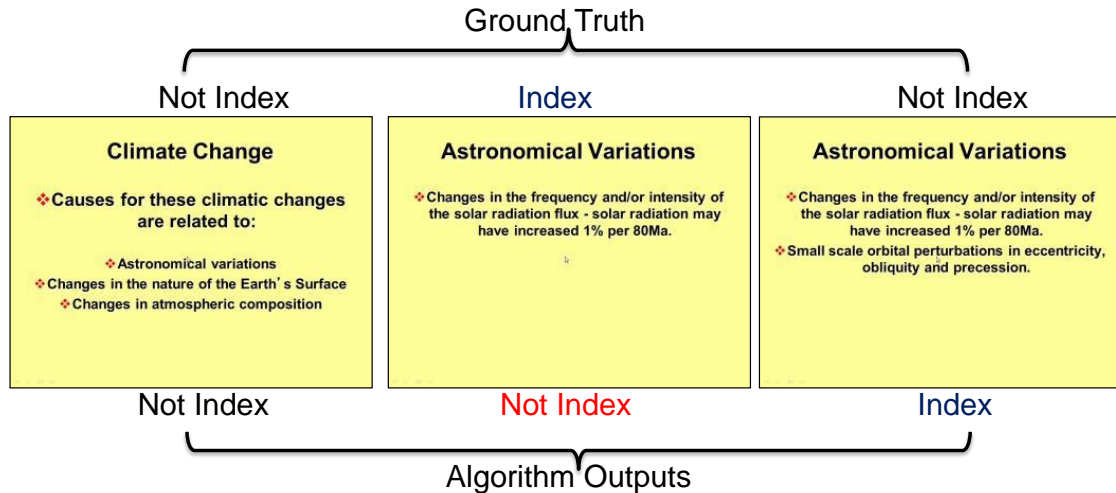


Figure 10.5: Animation and slide transition effect: next slide adds an additional line to previous one

10.1.4 Slide Without Enough Text for Topic Information

The title slide or the slides with the title of the following may not have enough topic information due to relatively small amount of text in it. This could cause false negatives by the merging of title slide to the previous topic, since the text may not be enough or may not completely represent the topic information that follows. Figure 10.6, shows an example of this type of error. The second slide is a start of a new topic but it is merged to the previous slide because of lacking of enough text information to compare.

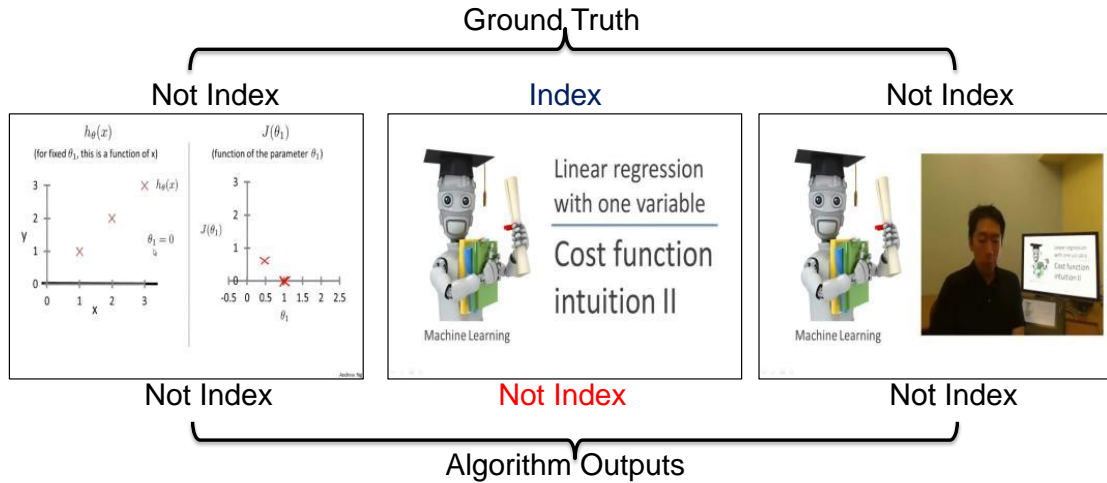


Figure 10.6: Slides have only images or have very low amount of text do not provide enough information for algorithms to correctly find index points

10.2 Other Common Errors

In this section, some of the other common errors are listed and depicted by an example. These errors are listed separately from the previous list, because in contrast to the previous errors, they were detected by some algorithms. In some cases, index points are missed by a text based algorithm; in some cases they are missed by a machine learning algorithm.

10.2.1 Slide Revisit

Occasionally the instructor may revisit a previous slide during the presentation, mostly for the clarification of that slide. Revisiting a slide or a previous topic causes a break in the similarity between the segments. When such break appears in the video, the next segment could be considered as a new topic although it is part of an

early discussion, due to very low similarity to the previous topic, thus resulting in an index point causing false positive. Conversely, in rare cases, merging the segments or topics in between the revisited and the previous topic to the previous topic itself cause false negatives. This is because, the presence of the revisited slide can cause higher similarity bias to the previous side or previous topic segment thus merging the in-between segment as shown in Figure 10.7. The instructor may not consider the revisit of slide as a topic change and prefer to ignore the revisit. However, the indexing algorithm could interpret this wrongly, thus causing errors.

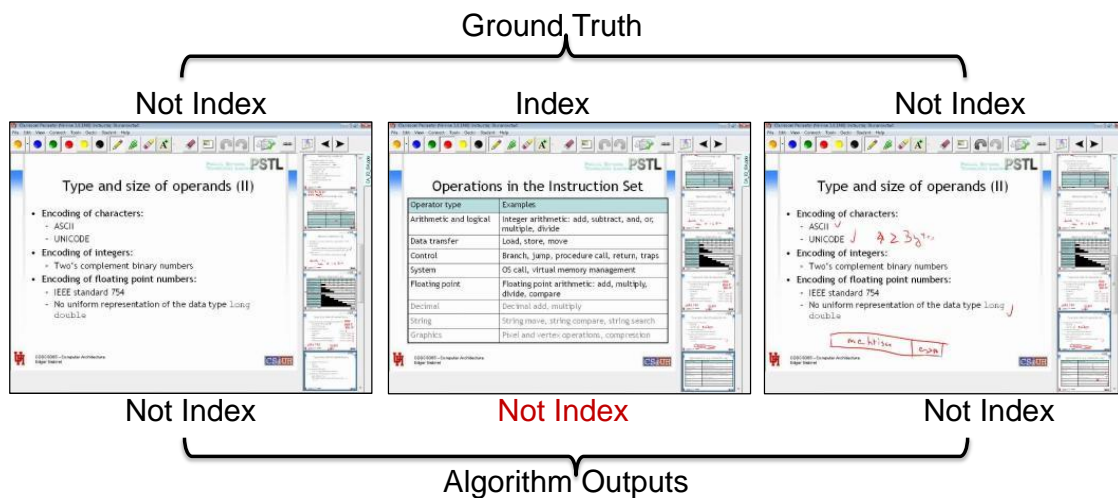


Figure 10.7: Going back to previous slides in lecture organization causes a break in the similarity between the segments

10.2.2 OCR Error and Hand Writings

When the instructor uses hand writings and draws images to explain the concept, OCR will not detect the text correctly which will cause new unrelated words to appear in the text. These new words may be considered as new topic by algorithms

since the similarities of consecutive frames will be different. Figure 10.8 shows an example of OCR error causing false positive: all the segments belong to the same slide and none of them marked as index point by instructor, but the second segment is found as index point by the algorithm.

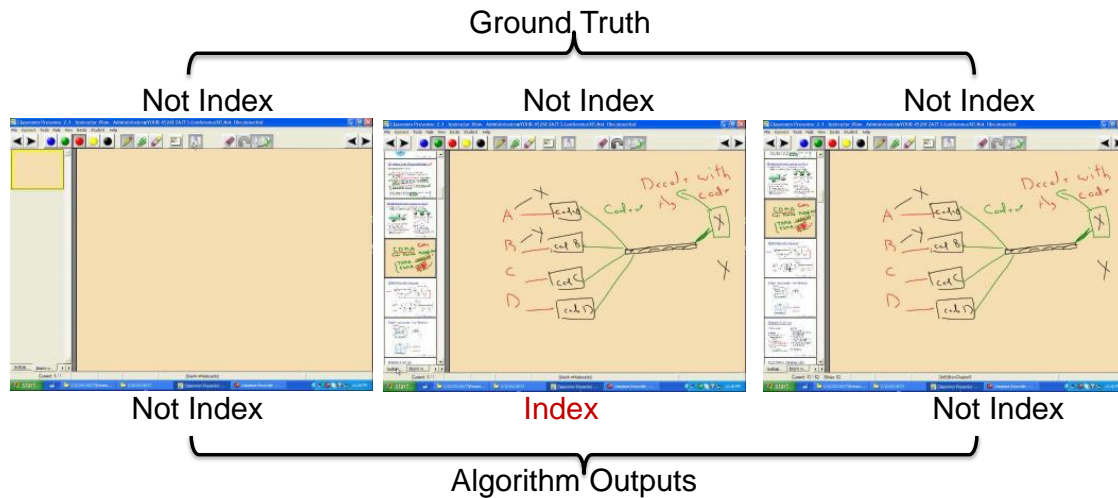


Figure 10.8: OCR errors on hand writing: false detections mislead the algorithm to treat same slide as different slide

10.2.3 Image Captions or Irrelevant Text

Sometimes there are some text which is irrelevant to the topic discussed in the class. When the instructor navigates in the folders of PC to load the presentation or when the class refers to a website and the instructor browses the websites, OCR tools will detect some texts which are not really related to topic discussed in the slides.

Having irrelevant text also happens with the captions of images in slides. As shown in Figure 10.9, the image caption text detected by OCR is correct but it has

no relation to the topic. Consequently, algorithms will choose arbitrary index points based on the similarities on irrelevant text.

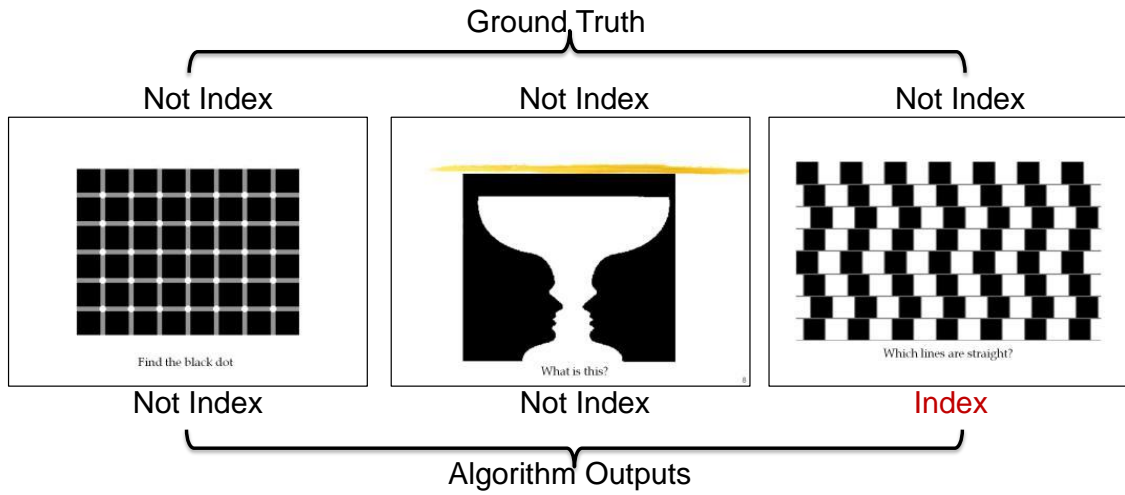


Figure 10.9: Texts which are irrelevant to the topic in image captions cause algorithms to produce false index point

Chapter 11

Conclusion

Lecture video indexing has a significant impact on accessing the content of interest. However, processing the video data is highly challenging, thus making automatic lecture video indexing a non-trivial task. This thesis work developed and evaluated the accuracy of text-based indexing algorithms and machine learning algorithms for automated, topic-based segmentation of lecture videos. A basic text-based indexing algorithm is introduced, variations of which are developed and evaluated. Dataset for machine learning is created with 406 features. The error analysis of outputs of the text-based approach and machine learning approach determined the probable causes of errors. The text-based indexing algorithm proved to be more effective in topic-based indexing compared to non-text-based method. By processing hundreds of features, machine learning significantly improved text-based indexing approaches.

11.1 Summary of Key Contributions

Our primary contributions are showing how to use the text information in video to divide lecture videos into different topic segments (1) by various text-based indexing algorithms and (2) by using machine learning algorithms. This includes extracting images from video and finding transition points, and extracting text by Optical Character Recognition(OCR) from video images with image enhancements. The following is a summary of our key contributions and novelties.

- Search Accuracy in OCR text improved from 91% to 97% by applying image enhancements prior to OCR engines.
- Text-based indexing algorithms are developed and they provided significant improvement over non-text-based approach and indexing with machine learning provided approximately
- The procedure of using the state of the art machine learning algorithms for video indexing is introduced and with RandomForest algorithm indexing accuracy is reached upto approximately 80% on average.
- To date, no systematic investigation has revealed what features are important to decide topic change in a lecture video. In this thesis, we have discovered that along with frequency of words, frequency of n-grams, number of first time words that appear in a video, words having large font size provides valuable information for video segmentation by topic.
- Error analysis for automated indexing is investigated and it is revealed that

slides having low amount of text, the consecutive slides having very similar topics, slides having outline, and slides loaded with animations make it difficult to detect topic change for automated video indexing.

- We have designed and developed the ICS Video framework which have been currently used by dozens of courses and reported positive feedback from students.

11.2 Future Work

Text-based indexing algorithms are using only text similarity by calculating the angle of frequency of words and slide duration which were not sufficient to define topic change. Machine learning experiment shows that there are more than 50 features important for finding lecture topic transition. In this regard, there are two approaches that can be followed to improve text-based indexing algorithms. One is to change the algorithms such that they do not only use cosine angle of frequency of words for decision of topic boundary but they also use all of the important features. The second way is to include these important features (such as the first time words that appear in a video, words having large font size and ngrams) in the current cosine angle of frequency of words vectors. But more weight should be given to these features because they have more value for deciding topic change than regular words.

Both text-based indexing algorithms and machine learning algorithms are required to be given a number of index points as an input parameter. This number is currently defined based on the duration of video, and currently set to index point per

6 minutes. This approach is practical for user interface such as Indexed Captioned Searchable Video Player. But statistics in previous chapter show that each video has a different number of index points per hour. Thus, any number of index points given per minutes will not work for all videos. Algorithms to find the number of index points in a video can be developed to provide the estimate or the exact number of index point to text-based indexing algorithms and machine learning indexing.

The current dataset is a diverse set of 25 lecture videos from different professors. Dataset creation for machine learning, processing each video and asking for professors to mark the index points is a time consuming task but it has significant value for machine learning algorithms. The more example there are for machine learning to learn, the more accuracy will be provided. Thus, providing more dataset to machine learning algorithms will increase the detection rate of index points. Furthermore, the assessment of generalization accuracy will be more reliable.

Unsupervised machine learning methods, specifically hierarchical clustering algorithms, are not investigated in this thesis. Those algorithms can be used to detect the index points or they can only be used to predict the number of index points.

One of the common errors for automated video indexing as listed in previous chapter is slides having not enough text or slides having only images. This challenge can be overcome by processing the speech text. Even though the Automated Speech Recognition tools have relatively poor recognition rates in contrast with Optical Character Recognition Tools, the speech text might be beneficial to video indexing algorithms.

Bibliography

- [1] ABOWD, G. D. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal* 38 (2000), 508–530.
- [2] ADCOCK, J., COOPER, M., DENOUE, L., PIRSIAVASH, H., AND ROWE, L. A. Talkminer: a lecture webcast search engine. In *Proceedings of the international conference on Multimedia* (New York, NY, USA, 2010), MM '10, ACM, pp. 241–250.
- [3] AHANGER, G., AND LITTLE, T. D. A survey of technologies for parsing and indexing digital video1. *Journal of Visual Communication and Image Representation* 7, 1 (1996), 28 – 43.
- [4] ALPAYDIN, E. *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.
- [5] ARMAN, F., HSU, A., AND CHIU, M.-Y. Image processing on compressed data for large video databases. In *MULTIMEDIA '93: Proceedings of the first ACM international conference on Multimedia* (New York, NY, USA, 1993), ACM Press, pp. 267–272.
- [6] BARKER, L., SUBHLOK, J., AND TUNA, T. Student perceptions of indexed, searchable videos of faculty lectures. In *Proceedings of the 44th Annual Frontiers in Education Conference* (Madrid, Spain, 2014).
- [7] BATT, G. Efficient automatic indexing of lecture videos. Master's thesis, University of Houston, 2010.
- [8] BIANCHI, M. Automatic video production of lectures using an intelligent and aware environment. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia* (New York, NY, USA, 2004), MUM '04, ACM, pp. 117–123.

- [9] BREIMAN, L. Bagging predictors. *Machine Learning* 24, 2 (1996), 123–140.
- [10] BREIMAN, L. Random forests. *Machine Learning* 45, 1 (2001), 5–32.
- [11] CAMASTRA, F., AND VINCIARELLI, A. Video segmentation and keyframe extraction. In *Machine Learning for Audio, Image and Video Analysis* (2008), Advanced Information and Knowledge Processing, Springer London, pp. 413–430.
- [12] CHAISORN, L., MANDERS, C., AND RAHARDJA, S. Video retrieval - evolution of video segmentation, indexing and search. *Computer Science and Information Technology* (2009), 16–20.
- [13] CHOI, F. Y. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference* (Stroudsburg, PA, USA, 2000), NAACL 2000, Association for Computational Linguistics, pp. 26–33.
- [14] DAVIS, M., KING, S., GOOD, N., AND SARVAS, R. From context to content: leveraging context to infer media metadata. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia* (New York, NY, USA, 2004), ACM Press, pp. 188–195.
- [15] DESHPANDE, R., TUNA, T., SUBHLOK, J., AND BARKER, L. A crowdsourcing caption editor for educational videos. In *Proceedings of the 44th Annual Frontiers in Education Conference* (Madrid, Spain, 2014).
- [16] FISHER, R., PERKINS, S., WALKER, A., AND WOLFART, E. Hypermedia image processing reference. *website: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm>* (2003).
- [17] FREUND, Y., AND SCHAPIRE, R. E. Experiments with a new boosting algorithm. In *In proceedings of the thirteenth international conference on machine Learning* (1996), Morgan Kaufmann, pp. 148–156.
- [18] HABERDAR, H., AND SHAH, S. K. Video synchronization as one-class learning. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand* (2012), ACM, pp. 469–474.
- [19] HABERDAR, H., AND SHAH, S. K. Change detection in dynamic scenes using local adaptive transform. In *British Machine Vision Conference 2013* (2013), BMVA Press, pp. 6–1.

- [20] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18.
- [21] HAMPAPUR, A., JAIN, R., AND WEYMOUTH, T. E. Production model based digital video segmentation. *Multimedia Tools Appl.* 1, 1 (1995), 9–46.
- [22] HANGER, A. A survey of technologies for parsing and indexing digital video. Technical Report TR-11-01-95, Boston University, 1995.
- [23] H.D., W., KANADE, T., AND SMITH, M.A. AND STEVENS, S. Intelligent access to digital video: Informedia project, 1996.
- [24] HEARST, M. A. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.* 23, 1 (Mar. 1997), 33–64.
- [25] HUANG, A. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZC-SRSC2008)*, Christchurch, New Zealand (2008), pp. 49–56.
- [26] JOUKOV, N., AND CHIUEH, T. Lectern ii: A multimedia lecture capturing and editing system. *International Conference on Multimedia and Expo 2* (2003), 681–684.
- [27] LAPATA, M. Automatic evaluation of information ordering: Kendall’s tau. *Comput. Linguist.* 32, 4 (Dec. 2006), 471–484.
- [28] LI, J. X. Automatic indexing of classroom lecture videos. Master’s thesis, University of Houston, 2008.
- [29] LIENHART, R., AND EFFELSBURG, W. Automatic text segmentation and text recognition for video indexing. *ACM/SPRINGER MULTIMEDIA SYSTEMS* 8 (1998), 69–81.
- [30] LIN, M., NUNAMAKER, J. J. F., CHAU, M., AND CHEN, H. Segmentation of lecture videos based on text: a method combining multiple linguistic features. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on* (2004), IEEE, pp. 9–pp.
- [31] MA, D., XIE, B., AND AGAM, G. A machine learning based lecture video segmentation and indexing algorithm. *Proc. SPIE 9021* (2013), 90210V–90210V–8.
- [32] MA, W.-H., LEE, Y.-J., DU, D.-C., AND MCCAHERILL, M. Video-based hypermedia for education-on-demand. *MultiMedia, IEEE* 5, 1 (1998), 72–83.

- [33] MA, W.-H., LEE, Y.-J., DU, D. H. C., AND McCAHILL, M. P. Video-based hypermedia for education-on-demand. In *Proceedings of the fourth ACM international conference on Multimedia* (New York, NY, USA, 1996), MULTIMEDIA '96, ACM, pp. 449–450.
- [34] MANNING, C. D., AND SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.
- [35] MERLER, M., AND KENDER, J. R. Semantic keyword extraction via adaptive text binarization of unstructured unsourced video. In *Proceedings of the 16th IEEE international conference on Image processing* (Piscataway, NJ, USA, 2009), ICIP'09, IEEE Press, pp. 261–264.
- [36] MO, H., YAMAGISHI, F., IDE, I., KATAYAMA, N., SATOH, S., AND SAKAUCHI, M. Key image extraction from a news video archive for visualizing its semantic structure. In *PCM (1)* (2004), pp. 650–667.
- [37] NAGASAKA, A., AND TANAKA, Y. Automatic video indexing and full-video search for object appearances. In *Proceedings of the IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems II* (1992), North-Holland Publishing Co., pp. 113–127.
- [38] NGO, C.-W., WANG, F., AND PONG, T.-C. Structuring lecture videos for distance learning applications. In *Multimedia Software Engineering, 2003. Proceedings. Fifth International Symposium on* (Dec 2003), pp. 215–222.
- [39] OTSUJI, K., AND TONOMURA, Y. Projection detecting filter for video cut detection. In *MULTIMEDIA '93: Proceedings of the first ACM international conference on Multimedia* (New York, NY, USA, 1993), ACM Press, pp. 251–257.
- [40] PERCANNELLA, G., SORRENTINO, D., AND VENTO, M. Automatic indexing of news videos through text classification techniques. In *Pattern Recognition and Image Analysis*, S. Singh, M. Singh, C. Apte, and P. Perner, Eds., vol. 3687 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 512–521.
- [41] REED, J. W., JIAO, Y., POTOK, T. E., KLUMP, B. A., ELMORE, M. T., AND HURSON, A. R. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *Proceedings of the 5th International Conference on Machine Learning and Applications* (Washington, DC, USA, 2006), ICMLA '06, IEEE Computer Society, pp. 258–263.

- [42] REPP, S., M. C. Semantic indexing for recorded educational lecture videos. *Fourth IEEE International Conference on Pervasive Computing and Communications Workshops* (2006), 240–245.
- [43] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3, 3 (July 1959), 210–229.
- [44] SHRIVAKSHAN, G., AND CHANDRASEKAR, C. A comparison of various edge detection techniques used in image processing. *IJCSI International Journal of Computer Science Issues* 9, 5 (2012), 269–276.
- [45] SUBHLOK, J., JOHNSON, O., SUBRAMANIAM, V., VILALTA, R., AND YUN, C. Tablet pc video based hybrid coursework in computer science: Report from a pilot project.
- [46] TOBAGI, F. Distance learning with digital video. *MultiMedia, IEEE* 2, 1 (1995), 90–93.
- [47] TUNA, T. Search in classroom videos with optical character recognition for virtual learning. Master’s thesis, University of Houston, 2010.
- [48] TUNA, T., SUBHLOK, J., AND SHAH, S. Indexing and keyword search to ease navigation in lecture videos. In *Applied Imagery Pattern Recognition Workshop (AIPR), 2011 IEEE* (Oct 2011), pp. 1–8.
- [49] TUNA, T., VARGHESE, V., SUBHLOK, J., JOHNSON, O., BARKER, L., AND SHAH, S. Development and evaluation of indexed captioned searchable videos for stem coursework. *SIGCSE ’12 Proceedings of the 43rd ACM technical symposium on Computer Science Education* (2012), 129–134.
- [50] TURNEY, P. D. Types of cost in inductive concept learning. In *Proceedings of the ICML’2000 Workshop on Cost-Sensitive Learning* (2003).
- [51] WOLPERT, D. H. The supervised learning no-free-lunch theorems. In *Proc. 6th Online World Conference on Soft Computing in Industrial Applications* (2001), 25–42.
- [52] YANG, Y., AND PEDERSEN, J. O. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning* (San Francisco, CA, USA, 1997), ICML ’97, Morgan Kaufmann Publishers Inc., pp. 412–420.